Expertise
and insight
for the future

Mai Vu

Introduction to Graph Algorithm in Network Routing

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Academic English and Project Communication Skills

Literature Review

24th September 2019

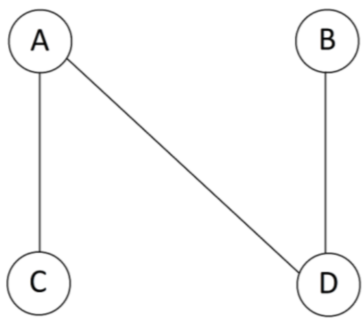
Contents

1 Overview

The topic of this literature review is to briefly introduce the use of graph theory in network routing. This topic fascinates me because I find myself wondering about the graph real-life applications. Graph theory is not only used in internet message routing but also used in other various tasks. For example, logistics, navigation, genome reading, social network analysis, etc. Therefore, learning about graph theory is useful and also important regardless of careers. The goal of this literature review is to give a background of graph theory (graph definition, types of graph) and to introduce graph algorithms in routing.

2 Graph Theory Background

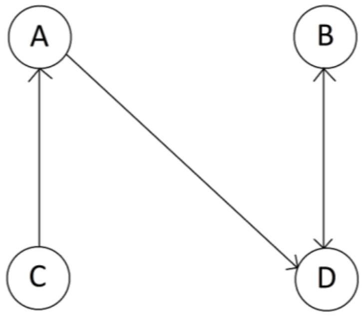
A graph is a set of vertices (also can be called points or nodes) and a set of edges or links connecting 2 nodes [1, 2]. Figures below illustrate some examples of graphs in the 2-dimension plane.



to	A	B	C	D
A	1	0	1	1
B	0	1	0	1
C	1	0	1	0
D	1	1	0	1

Figure 1. Undirected graph G_1 on 2-dimension plane with its matrix.

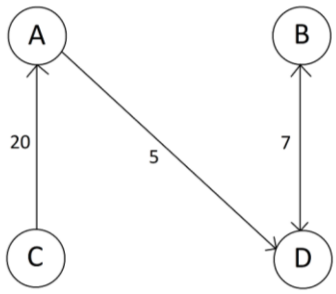
According to figure 1, there are 4 nodes and 3 edges. The matrix next to graph G_1 represents it. If there is a link between 2 nodes, the cell that corresponds for the connection of that 2 nodes labels 1; if not, it labels 0. Additionally, the cell for the link between a node itself also labels 1.



	A	B	C	D
A	1	0	0	1
B	0	1	0	1
C	1	0	1	0
D	0	1	0	1

Figure 2. Directed graph G_2 on 2-dimension plane with its matrix.

Graph G_2 in figure 2 also has 4 nodes and 3 edges. Its matrix is also similar to that of G_1. The main difference is that G_2 is a directed graph, meaning the edges or paths have directions [1, 25]. For example, there is a path from C to A, but not otherwise. Additionally, the undirected graph G_1 can be also considered as a directed graph which has 2-direction paths.



	A	B	C	D
A	0	∞	∞	5
B	∞	0	∞	7
C	20	∞	0	∞
D	∞	7	∞	0

Figure 3. Weighted graph G_3 on 2-dimension plane with its matrix.

Edges can also hold values as in figure 3; therefore, graph G_3 is called a weighted graph [2, 49]. Depending on the problem, values can represent for the time, the distance or the cost from a node to another. Different from the other 2 matrices, which only show whether there is a path or not, the G_3 matrix shows the value of the paths. If there is no path, the value is set to infinite. The value for the path between a node itself is set to 0.

There are more types of graphs besides those 3 mentioned above. There are also different methods to represent graphs [1, 36]. In the next section, weighted undirected graphs are used as examples for the testing of graph algorithms.

3 Network Routing

3.1 Introduction

The process of transporting messages within a network or from a network to another operated by the Network layer is called network routing [3, 1]. There are 2 common types of network routing: static routing and dynamic routing. In static routing, routes are added manually by administrators [4, 2]; therefore, it does not require automatic updating routes algorithms as dynamic routing.

Dynamic routing, or adaptive routing, applies graph algorithms to adjust routes according to changes in the network topology as well as network traffic [4, 5]. Distance-vector routing algorithm and link-state routing algorithm are the 2 important routing algorithm used by dynamic routing [5, 243]. Nodes in distance-vector routing do not know all the nodes [5, 246] as in link-state routing [5, 253]; however, they know nodes that directly attached to and the direction (or the next hop) for data to be sent.

3.2 Distance-vector Routing Algorithm

As mentioned, nodes in distance-vector routing have no information about the cost of routes and data is sent along the direction of the least-cost path found by an algorithm called Bellman-Ford [5, 243]. The below example is examined for a better understanding.

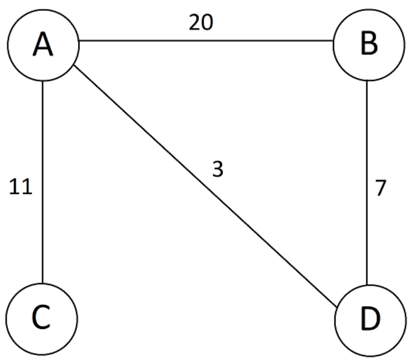


Figure 4. Graph G_4 – a simple 4-router network.

According to figure 4, there are 4 routers or nodes named A, B, C, D and the weighted edges representing the amount of time for a successful transmission or the distance between 2 nodes. The task is to complete routing tables for each node. The routing table at node S tells a final destination F, the next hop to get to F and the total cost from S to F. Figure 5 shows the process of constructing the tables.

At t = 1

A's routing table

Des	Next	Cost
B	B	20
C	C	11
D	D	3

B's routing table

Des	Next	Cost
A	A	20
D	D	7

C's routing table

Des	Next	Cost
A	A	11

D's routing table

Des	Next	Cost
A	A	3
B	B	7

At t = 2

A's routing table

Des	Next	Cost
B	D	10
C	C	11
D	D	3

B's routing table

Des	Next	Cost
A	D	10
C	A	31
D	D	7

C's routing table

Des	Next	Cost
A	A	11
B	A	31
D	A	14

D's routing table

Des	Next	Cost
A	A	3
B	B	7
C	A	14

At t = 3

A's routing table

Des	Next	Cost
B	D	10
C	C	11
D	D	3

B's routing table

Des	Next	Cost
A	D	10
C	D	21
D	D	7

C's routing table

Des	Next	Cost
A	A	11
B	D	21
D	A	14

D's routing table

Des	Next	Cost
A	A	3
B	B	7
C	A	14

Figure 5. G_4 routing tables using Bellman-Ford algorithm

At the beginning (t = 1), a node's routing table only contains information of its next-door-neighbors [5, 244]. After that (t = 2), nodes send and receive routing tables to/from the directly connected neighbors [5, 245]. Each node then compares its routing table and received tables. If a node finds a better path or a new destination, its table is updated. For example, because A has directed paths to all 3 nodes, A gets tables from all B, C, D. Node A notices it can get to B through D with a cost is a total cost to get to D from A and from D to B, which is 3 + 7 = 10. A updates its routing table. If A received data which is intended for B, A sends the data to the next hop D. Additionally, because there are new addresses have not been in the original tables, routing tables at node B, C and D also get updated. The task is carried on until there are no further updates. The final routing tables are shown at t = 3.

3.3 Link-state Routing Algorithm

In link-state routing, all nodes first flood the massage about its directed neighbors to the whole network [2, 253]. Then, each node separately builds up a graph using the information it received and applied Dijkstra's algorithm to find shortest paths to other nodes [5, 256].

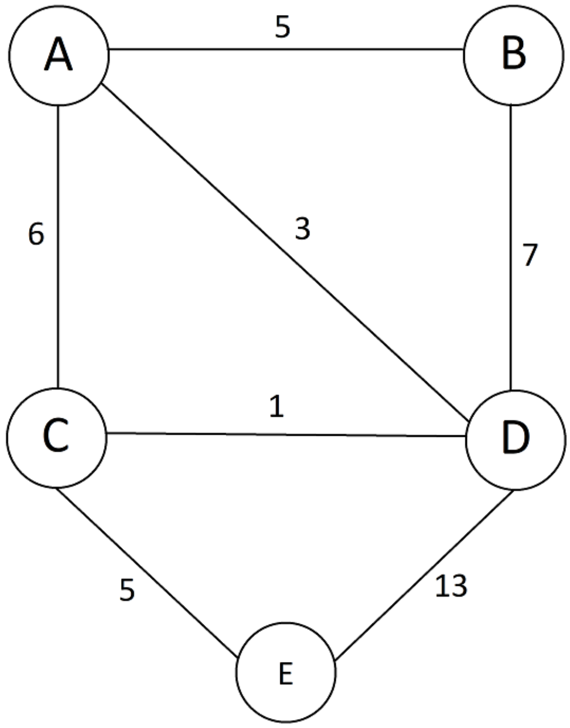


Figure 6. Graph G_5 - a simple 5-router network.

Using graph G_5 from figure 6, figure 7 illustrates an example of the Dijkstra process. As mentioned earlier, Dijkstra's algorithm is performed independently at every node. This example uses node A as the starting point. The task is still to fully develop the routing table at each node. The routing table in this algorithms is similar to that of distance-vector one, but instead of showing the next hop, it shows the previous hop of the destination. Dark brown color indicates the node has not been visited, while light brown means the node is currently visiting. White color tells the node has been visited, meaning the path to it is the least-cost path and the node does not need to be visited again.

t = 1

Des	Pre	Cost
B	A	5
C	A	6
D	A	3
E	-	∞

t = 2

Des	Pre	Cost
B	A	5
C	D	4
D	A	3
E	D	16

t = 3

Des	Pre	Cost
B	A	5
C	D	4
D	A	3
E	C	9

t = 4

Des	Pre	Cost
B	A	5
C	D	4
D	A	3
E	C	9

t = 5

Des	Pre	Cost
B	A	5
C	D	4
D	A	3
E	C	9

Figure 7. G_5 routing table at node A using Dijkstra's algorithm

At the beginning (t = 1), node A knows the number of nodes. It starts to full the routing table with nodes. As there are directed paths from A to B, C, and D, those costs and the previous-hop A are added; however, A and E are not neighbors, so the cost to get to E is infinite and no previous hop. A then picks the hop that the cost is the least which is hop D to continue the next stage. The choice is colored light brown.

At t = 2, all directed unvisited neighbors of D (B, C and E) are taken into consideration. It takes 3 + 7 = 10 costs to get to B through D; comparing with 5 costs to B from A, the old value is kept. However, A can get to C through D with only 3 + 1 = 4 costs, so A updates the new path: C's previous hop is D and the total cost is 4. As E is also D's neighbor, A adds E's previous hop is D with the cost is the total cost from A to D and from D to E. After all costs are updated, A marks hop D as visited by white color and again chooses the unvisited hop that has the least-cost, signs it with light brow and begins a new stage. Dijkstra's algorithm finishes when all nodes are visited as at t = 5.

4 Summary

This literature review discussed a general view of graph and graph algorithms in network routing; however, did not cover the difference between those 2 common routing protocol or their improvement in the distributed environment. This literature review simply gave a glance into the routing; therefore, the algorithms or the routing tables are written in an easier, more understandable form. For further reading about graphs and graphs in network routing, documents in the reference are highly recommended. Thanks to the implement of graph theory in routing, billions of data from all over the world can be transmitted faster to serve nowadays demand.

Graphs are an interesting subject. The study of graphs for me is also the study of relationships or real network connections. I believe there are still plenty of possibilities for more graph researches and for applying graphs in other aspects of life.

References

1 Dieter Jungnickel. Graphs, Networks and Algorithms (Fourth Edition). Springer-Verlag Berlin Heidelberg; 2013.
URL: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Graphs%2C%20Networks%20and%20Algorithms%20%284th%20ed.%29%20%5BJungnickel%202012-11-09%5D.pdf
Accessed 14th September 2019.

2 Santanu Saha Ray. Graph Theory with Algorithms and its Applications in Applied Science and Technology. Springer India; 2013.
URL: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Graph%20Theory%20with%20Algorithms%20and%20its%20Applications_%20In%20Applied%20Science%20and%20Technology%20%5BRay%202012-11-02%5D.pdf
Accessed 14th September 2019.

3 Microsoft. Routing Concepts. 2008.
URL: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc957843(v=technet.10)
Accessed 18th September 2019.

4 Microsoft. Static and Dynamic Routers. 2008.
URL: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc957844(v=technet.10)?redirectedfrom=MSDN
Accessed 19th September 2019.

5 Bruce S. Davie and Larry L. Peterson. Computer Networks: A Systems Approach. 2012.
URL:
https://cseweb.ucsd.edu/~gmporter/classes/wi19/cse124/courseoverview/compnetworks.pdf
Accessed 20th September 2019.