Report of Project3

Routing Algorithms via Python

Qiong Yang

1 Problem Description

Given Shanghai Metro Map, we need to implement two routing algorithms(Link-State Routing and Distance-Vector Routing) to find the shortest path between arbitary two stations. The input and output can be described as follows:

- Input: Shanghai Metro Map and arbitrary two stations(for example:from Zhangjiang Hi-tech Park to Zhaojiangbang Rd)
- Output:the least path or the least switching times with delay as low as possible

2 Problem Analysis

1. Target:Find the shortest path between two arbitrary station.
2. Algorithm:Link-State Routing Algorithm & Distance-Vector Algorithm
3. Tools:Python2.7.11

According to the above analysis,we need to complete two steps.
Step1: Look for the Shanghai Metro Map and transform it into graph struct via python.
Step2: Write two kinds of algorithm according to the graph struct

3 Process Realization

3.1 Construct Topology

Figure 1: Shanghai Metro Map

3.2 Algorithm Realizing

After constructing the topology, we need add the two kinds of algorithm into pox controller so that we can achieve the target of finding the shortest path.

Firstly, I will give the algorithm process.

3.2.1 Link-State Algorithm(Dijkstra)

The link-state protocol is performed by every switching node in the network (i.e., nodes that are prepared to forward packets; in the Internet, these are called routers). The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible

destination in the network. The collection of best paths will then form the node's routing table.
Link-state routing requires that all routers know about the paths reachable by all other routers in the network. Link-state information is flooded throughout the link-state domain (an area in OSPF or IS-IS) to ensure all routers posses a synchronized copy of the area's link-state database. From this common database, each router constructs its own relative shortest-path tree, with itself as the root, for all known routes.
The Dijkstra algorithm solves the single-source shortest path problem for a graph G with non-negative edge weights, producing a shortest path tree. It is a greedy algorithm that starts at the source node, then it grows T and spans all nodes reachable from the source. Nodes are added to T in order of distance. The relaxation process is performed on outgoing edges of minimum-weight nodes. The total time is $O(ElogV)$.
The Algorithm steps are below.
Step1: Choose source node,S=v,U=other nodes.If node v connect with node u(in U),then ¡u,v¿ has weight.Otherwise the weight of ¡u,v¿ is inf.
Step2: Choose node k from U and the path between k and v is the shortest. Add the node k into S.
Step3: Take the node k as the updated middle node, update the distance between source node and nodes in U.If distance between v and u(contain k) is shorter than origin distance (not contain k), then update the distance between source and node u.
Step4: Repeat the step2 and step3 until all nodes in S.
By this algorithm, I can get the shortest path from Zhaojiabang Rd station to ZhangjiangHighTechPark station.

```
========== RESTART: C:\Users\IBM\Desktop\Link-State.py ==========
please input the sorce station: Zhaojiabang Rd
please input the end station: ZhangjiangHighTechPark
Dijkstra's shortest path is
The shortest path : ['Zhaojiabang Rd', 'Jianshan Rd', 'Dapuqiao', 'Madang Rd', 'LUjiabang Rd', 'Xiaonanmen',
'ShangchenRd', 'CenturyAvenue', 'ShanghaiScience&TechnologyMuseum', 'CenturyPark', 'LongyangRd', 'ZhangjiangH
ighTechPark']
The time cost is 28
```

Figure 2: The shortest path from Zhaojiabang Rd station to ZhangjiangHigh-TechPark station

I compare the result with Shanghai Metro Map and I find that the path is correct but the cost time is too little because I don't consider the switch time.

3.2.2 Distance-Vector Algorithm(Bellman-Ford)

Distance vector routing, in contrast, routing information is only exchanged between directly connected neighbors. This means a router knows from which neighbor a route was learned, but it does not know where that neighbor learned the route; a router can't see beyond its own neighbors. This aspect of distance vector routing is sometimes referred to as "routing by rumor." Measures like split horizon and poison reverse are employed to avoid routing loops.
The Bellman-Ford algorithm computes shortest paths from a single source to all of the other nodes in a weighted graph G in which some of the edge weights are negative. The algorithm relaxes all the edges and it does this $|V| - 1$ times. At the last stage, negative cycles detection is performed and their existence is reported. The algorithm runs in $O(V * E)$time.
By this algorithm, I can get the shortest path from Zhaojiabang Rd station to ZhangjiangHighTechPark station.

```
============== RESTART: C:\Users\IBM\Desktop\Distance-Vector.py ==============
please input the sorce station: Zhaojiabang Rd
please input the end station: ZhangjiangHighTechPark
'The Bellamn_ford's shortest path is
The shortest path : ['Zhaojiabang Rd', 'Jianshan Rd', 'Dapuqiao', 'Madang Rd', 'LUjiabang Rd', 'Xiaonanmen',
'ShangchenRd', 'CenturyAvenue', 'ShanghaiScience&TechnologyMuseum', 'CenturyPark', 'LongyangRd', 'ZhangjiangH
ighTechPark']
The time cost is 28
```

Figure 3: The shortest path from Zhaojiabang Rd station to ZhangjiangHigh-TechPark station

4 Conclusion

Via this project, i learn core of the two kinds of algorithm and differences.
The difference between link-state and distance-vector algorithm is that :
Link-State send routing information to all nodes on Internet but the routing information only contains the information related to themselves.
However, distance vector algorithm requires each router send all information to their neighbor. These information contains other nodes'routing information.
However, distance vector algorithm requires each router send all information to their neighbor. These information contains other nodes'routing information. The complete code can be seen on github.
`https://github.com/QiongYang/Routing-Algorithm-via-Python2.7.git`