ABSTRACT

Routing is one of the main methods of accessing the Internet. In this process, the assignment of IP addresses, the transmission of information, etc. cannot be done without the support of algorithms. A good routing algorithm is critical to the experience of information delivery. Based on this reposition, the project is crafted to help student practice evaluating the structure of routing and algorithms behind it.

INTRODUCTION

This project, focusing on building a simple routing system, requires students to have a basic understanding on Bellman-Ford distance vector algorithm, network coding, routing and python socket network programming. In modern society, information is becoming more and more important, and the routing has become a major technology for mass exchanging, transferring and storing information over a great number of nodes. The development of routers is rapid, new generation of smart routers like Amazon echo and Google home have combined web services and other smart device calls with human-computer interaction to become an important part of the smart home system in the IOT era [1].

Methodology

See comments in source code plz.

Implementation

The program is divided into two main parts

1. UDP sending part, used to send the current node information

The node information mainly contains current node, distance from current node to other nodes, next hop from current node to other nodes

Node processing

According to the current node name, get the distance file of the current node, node\_distance.json under test\_data

file, process the file and convert it to python-readable dict type (further processing is needed here1. if the current node contains its own loop information).

If the previous node contains information about its own loop, remove that information because the route will not reach itself. 2. For nodes that cannot be directly

For nodes that cannot be reached directly, add the node to the node information, with distance defined as 10000 and next hop defined as node)

Then create a UDP connection and broadcast (send) your node information to each node.

Then create a dead loop and keep updating the node\_output.json file, which holds the current node information. Until forced

exit.

2. the UDP receive part, which is used to receive information from other nodes asynchronously until the main thread exits.

This part is a coexisting word thread program with the main thread to continuously receive information from all nodes.

Once the node information is received, the bellman algorithm is done on the node information.

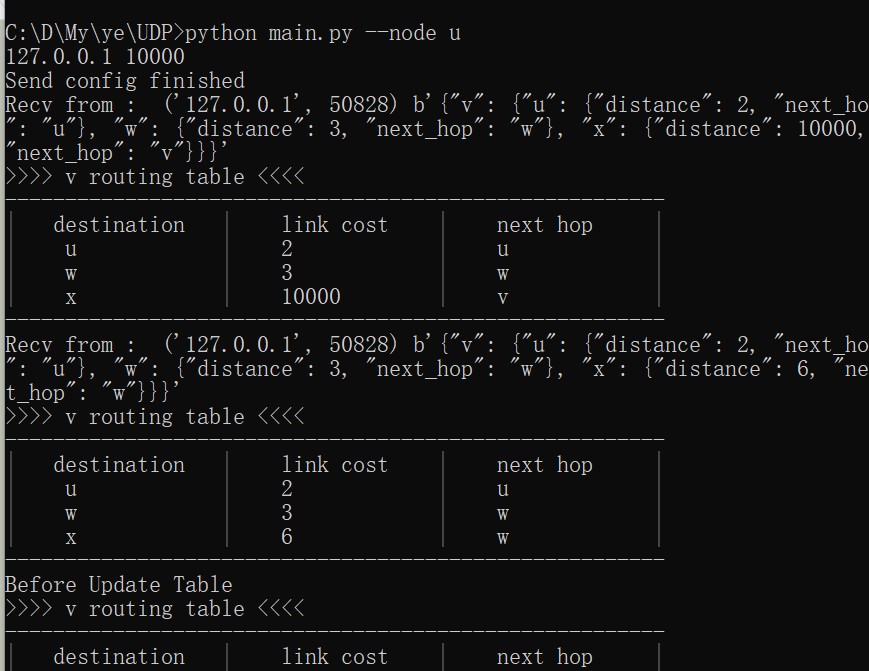
Detect the distance to a node saved by the current node (assumed to be a), and then add the distance to the current distance received plus the

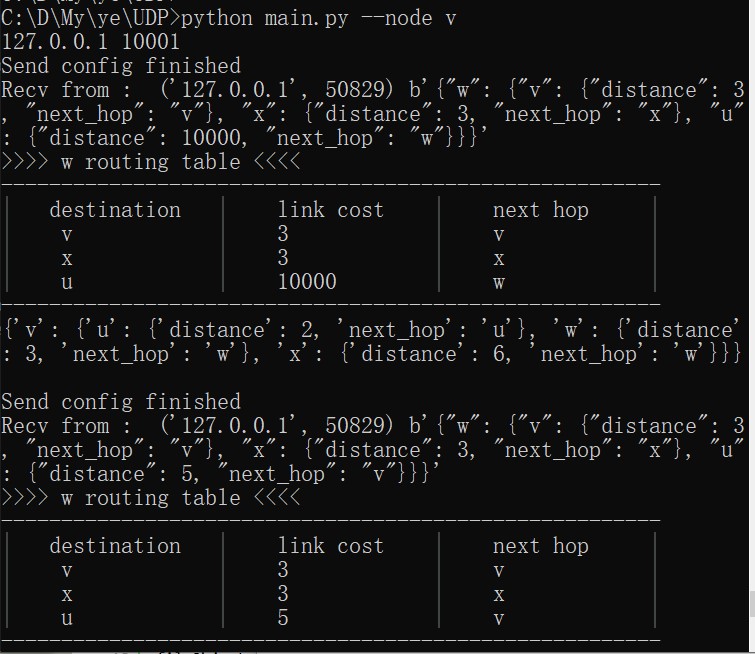
If a>b, then the distance is relaxed and the distance of the current node is updated to b. The new node information is broadcast to other nodes.

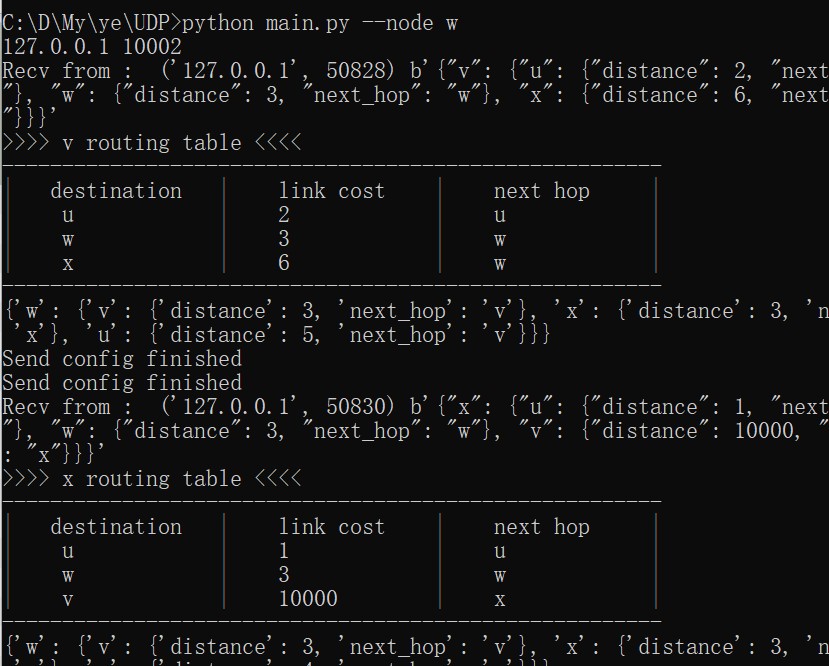
The new node information is broadcasted to other nodes.

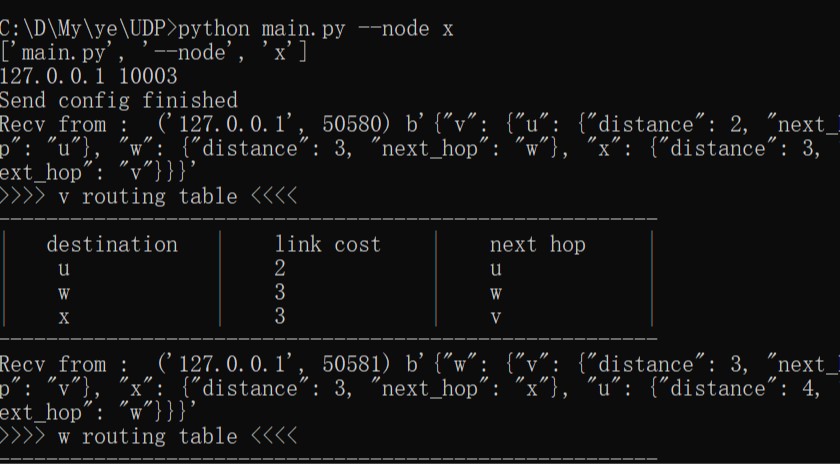
Testing and results

Fig below showed processing mechanics.







Conclusion

The project was successfully completed with majority of the tasks accomplished. No major issues occurre during coding.

Reference lists

[1] Danny Yuxing, Huang,Noah J Apthorpe, Frank Li, Gunes Acar ,Nick Greer Feamster

“IoT Inspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices at Scale” in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies Volume 4, Issue 2:* Association for Computing Machinery, New York, NYUnited States, June 2020

<https://dl.acm.org/doi/abs/10.1145/3397333>