**COE 865 Project**
**Design and Implement a Routing Control System for Inter-domain Routing**
*Due Date:* ***April 10, 2022***

In Internet, BGP is currently used for inter-domain routing. Border routers select inter-domain routes based on connectivity and local policies. In this project, you will design and implement a routing control system that helps select networks in the inter-domain routes based on a combination of parameters such as available bandwidth and cost.

## Overlay Network of Route Controllers

Figure 1 shows an interconnected network of several autonomous domains. A Route Controller (RC) is deployed in each of the four *controller* networks (autonomous systems), ASN100, ASN200, ASN300 and ASN400. The route controllers – RC1, RC2, RC3 and RC4 – form an overlay network by connecting with each other through UDP connections (thick lines). The controller networks are connected to each other and to other networks indicated by their AS numbers. For example, the controller network ASN 100 with RC1 is connected to ASN 10, ASN 20, ASN 200 and ASN 300. The route controllers periodically exchange information about the connections of their networks. The information received by a route controller is used to compute the optimal route from the controller network to all other networks.

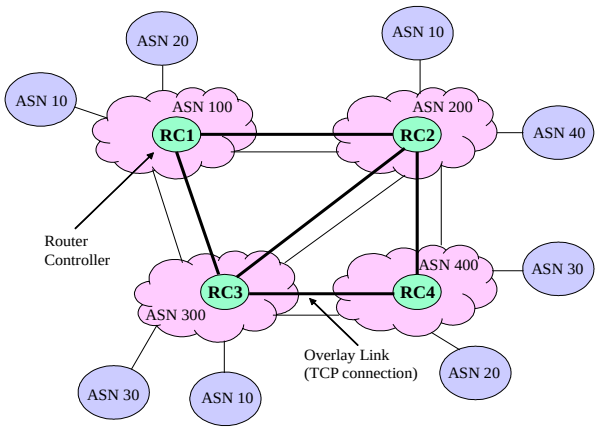


Figure 1: Route Control Overlay Network

## Design Goals

1. Define a composite cost function for path selection based on link capacity and the link rental cost given in dollar amount.
2. Develop and implement optimal path selection algorithm.

## Given

1. A Route Controller is deployed in ASN 100, ASN 200, ASN 300 and ASN400 networks.
2. A Route Controller selects *optimal* routes from its network to all other networks.
3. A route controller sends periodically Route Control Update (RCU) messages to the RC_PORT of each of its neighbor route controller every RCU_TIMER seconds.
4. The RCU message contains a set of 5-tuple – RCU message: {<RCID, LOCAL-ASN, DEST_ASN, Link Capacity, Link Cost}. The fields of an RCU message are defined below:
    - RCID: Route Controller ID

- LOCAL_ASN: AS number of the local controller network
- DEST_ASN: AS number of the destination network
- Link Capacity: the bandwidth of the link between LOCAL_ASN and DEST_ASN
- Link Cost: The dollar cost of renting the link between LOCAL_ASN and DEST_ASN, e.g. $100/Mbyte etc.

**Program**

- You should write a program to implement the above overlay network with your route control algorithm in Python (or C, C++, Java). The recommended programming language is Python using Python 3.5 or higher version. *It is much easier to develop this code in Python with a rich library of packages. Python has several options available for inter-process communication (IPC). You can use Multiprocess package to spawn multiple processes.*
- You can use any IPC mechanism, such as Pipes, Queues or Socket APIs. Alternatively, you can use messaging system such as MQTT (www.mqtt.org) or Rabbit MQ (www.rabitmq.com).
- Socket API: You can visit http://beej.us/guide/bgnet/ for on-line tutorial on socket programming. You should use loopback address, 127.0.0.1, and port number within the following interval (65000 < port numbers < 65535) at each node to run all of them on the same computer. You can either use TCP (preferrable for reliability reason) or UDP socket to establish with the peers for exchanging messages.
- Use following configuration parameters:
  - RCU_TIMER 180 seconds (3 minutes)
  - RC_PORT 1450 (if you are using socket)
- You can assume maximum number of networks 10.
- You should generate your own test topology by modifying the topology of Figure 1.
- You should test with four route controllers as shown in Figure 1.

**Input:** The input to the program is a *config* file. A sample config file is posted in D2L with the format details.

**Output:** Display on the screen the routing table the destination network and the path chosen for that. For example, DEST_ASN=40, PATH= {ASN 400, ASN 200, ASN 40}, BW, COST

**Submission**: You can work in a group of no more than three students. You should submit a tar file containing all program files (e.g. *.py), *config* files, Readme, and project report. The Readme file should include instructions how to run the program. The report must describe all the program files, major algorithms, message structures, test cases and the topologies you use for testing your program. You should also include analysis of your routing algorithm and how it can be used on the Internet. You should also identify in the report the contribution of each team member. Report will be evaluated not only on the technical content but also on the quality of presentation.

**Evaluation**

The whole team must participate in the test run after the due date. Each team member will be asked question related to their individual contributions.

| | |
|---|---|
| Test run | 30% |
| Program (code) | 30% |
| Report | 40% |