CPE400
Project Two
BGP Simulator

Members: Andy Singh, Ben Wendell, Byron Shure

Simulator Design

Our program is a simulation of the BGP protocol that includes BGP as well as iBGP. We begin by reading in a file that contains a full simulated network, including AS data and routers internal to each AS. Each autonomous system has an IP prefix associated with it. The routers internal to each AS contain ip addresses that fall within that prefix. The network, by default, contains three autonomous systems. Each AS contains 3 routers, though more can be added later, and each router contains 4 hosts. When we read in each router, we read in attribute information that includes: weight, local preference, MED, origin, and community. In addition, we read in the subnets that this router provides access to.

Once we have read in this data, we create a weighted list for each AS. From here, we add edges that correspond to internal paths in the AS, with weights derived from network characteristics. At the end of this initialization, we have 3 weighted graphs that each represent an independent AS. The vertices in these weighted graphs represent routers internal to the AS and the edges represent a cost between them, together, these vertices and edges represent the internal state of each AS.

After building our network and the individual AS's, we allow the user several options: they can view details about the network as a whole, about a specific AS, or a specific router contained within an AS. Viewing this information provides insight on how this dummy network is organized as well as providing additional helpful information explaining attributes that affect how BGP controls routing behavior.

In addition to viewing this information, they can add a new router, or change the link that connects the internal routers to the AS. Finally, the user is able to simulate sending a tcp segment, complete with a reliability check that will resend the segment if it is lost.

Network Layers

BGP is an application layer protocol that uses TCP to communicate with other routers to share available paths. While we don't simulate the actual TCP connection that is used to share this information between routers, the information is propagated through the use of our AS weighted graphs.

Additionally, we allow the user to add a router internal to an AS, or change the link from the internal AS to the outside world. When a user adds a router, it is added to the weighted graph and the user is able to view it like any other router, or use the hosts contained within as either source or destination for the TCP segment.

We also simulate the path-finding that occurs when a data packet is sent from a router within an AS to another router that may or may not be in another AS. This path finding involves ip addresses and the network layer. By simulating the sending and receiving of mock data, we are exposing details of the Application layer(sending application data), the transport layer (TCP), and the network layer (IP routing). During the sending of mock packets, we simulate the sending and receiving of each individual packet as well as each acknowledgement that should be received. If packets are lost, they are retransmitted in accordance with a reliable protocol. Our TCP simulation does not follow TCP exactly -- in order to detail the progress and provide a visualization -- we use something closer to stop-and-wait.

Java Visualization

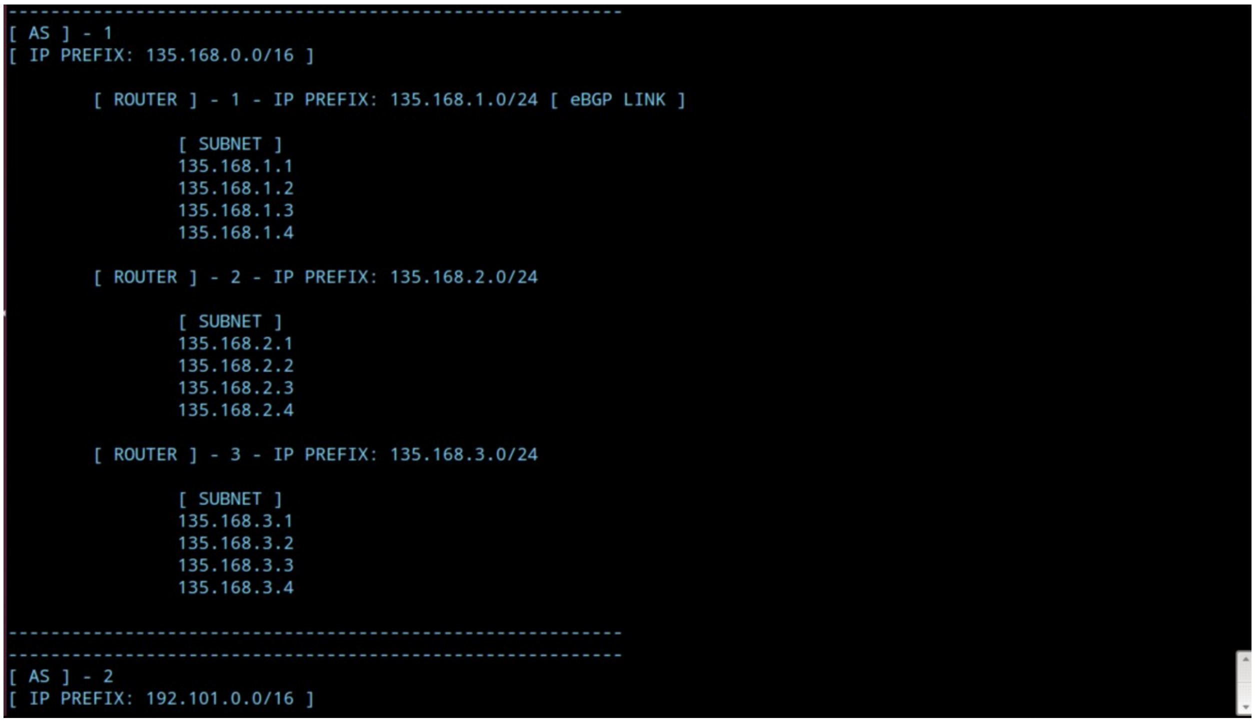
In addition to our console simulation of BGP routing, we have provided a java program that further illustrates BGP routing. This program uses a picture from one of your slides, and allows the user to select hosts to send data between. It shows pathfinding while simulating a packet being sent, which is resent if it is lost.

Design Documents

In addition to our commented source, we have provided Fig. 1 which is a simplified flow diagram detailing the basic operations permitted by our program. Fig. 2 gives a visual representation of the default network that is setup by the programs data.txt, it focuses on the three external AS’s, as well as the internals of AS1.

Screen Shots of Program

This is an image of AS 1. This AS has an IP prefix. Contained in the AS are three routers, which also have an IP prefix as well as their subnet. The other two ASes are similar with their own unique IP prefixes.



```
----------------------------------------------------------
[ AS ] - 1
[ IP PREFIX: 135.168.0.0/16 ]

        [ ROUTER ] - 1 - IP PREFIX: 135.168.1.0/24 [ eBGP LINK ]

                [ SUBNET ]
                135.168.1.1
                135.168.1.2
                135.168.1.3
                135.168.1.4

        [ ROUTER ] - 2 - IP PREFIX: 135.168.2.0/24

                [ SUBNET ]
                135.168.2.1
                135.168.2.2
                135.168.2.3
                135.168.2.4

        [ ROUTER ] - 3 - IP PREFIX: 135.168.3.0/24

                [ SUBNET ]
                135.168.3.1
                135.168.3.2
                135.168.3.3
                135.168.3.4

----------------------------------------------------------
----------------------------------------------------------
[ AS ] - 2
[ IP PREFIX: 192.101.0.0/16 ]
```

This shows the options that are allowed to update the network. You are allowed to update the eBGP link, which is used to talk to other ASes. You can choose another router within the AS to be the eBGP link.

```
[ UPDATE NETWORK ]

[ 1 ] UPDATE eBGP LINK
[ 2 ] ADD ROUTER TO AS
[ 3 ] MAIN MENU

PLEASE MAKE A SELECTION: 1
[ 1 ] - AS #1
[ 2 ] - AS #2
[ 3 ] - AS #3

PLEASE SELECT AN AS: 1

CURRENT eBGP ROUTER LINK: 135.168.1.0

ROUTER LIST :
1       135.168.1.0
2       135.168.2.0
3       135.168.3.0

PLEASE SELECT A ROUTER: 1

NEW eBGP ROUTER LINK:135.168.1.0
```

This shows the other network update option that allows you to “Start a new company” or add another route to an existing AS. The program requires you to enter BGP attributes when adding a new router that are used to determine the weight between the links of the internal structure of the AS. Once the router has been created it is linked with another router within the AS.

```
[ BGP ATTRIBUTES ]

PLEASE ENTER WEIGHT: 100

PLEASE ENTER LOCAL PREFERENCE: 12

PLEASE ENTER MED: 12

ORIGIN:
IBGP
EBGP
Incomplete

PLEASE ENTER ORIGIN:
Incomplete

COMMUNITY:
no-export
no-advertise
Internet

PLEASE ENTER COMMUNITY:
Internet

ROUTER LIST

1       135.168.1.0
2       135.168.2.0
3       135.168.3.0

SELECT ROUTER TO CREATE LINK WITH:
1
```

This shows when the program is going to send data through the network. It gets the reachability of the destination host & prepare to send the data.

```
OBTAINING SUBNET REACHABILITY FROM NEIGHBORING AS
THE MESSAGE: "CPE400" HAS BEEN BROKEN INTO: 6 PACKETS THAT WILL BE TRANSMITTED
SOURCE HOST: 135.168.1.1 STARTING A TCP CONNECTION WITH DESTINATION HOST: 235.155.2.3

TCP CONNECTION IS NOW OPEN

PREPARING TO SEND [ PACKET #1 - DATA: ( C ) ]
```

This shows the optimal path that was determined by the shortest path that was taken within the AS & linked by eBGPs. The source host sends the packet up to its source router, which then gets forward to the eBGP link. The router then sends it to the destination eBGP link of the other AS that then trickles down to the destination host.

```
                                                    [ ROUTE ]

                                           [ PRESS ENTER TO SIMULATE ]

[ SOURCE HOST ]     [ SOURCE iBGP ]     [ SOURCE eBGP ]     [  DEST eBGP  ]     [  DEST iBGP  ]     [  DEST HOST  ]
[ 135.168.1.1 ] --> [ 135.168.1.0 ] --> [ 135.168.4.0 ] --> [ 235.155.1.0 ] --> [ 235.155.2.0 ] --> [ 235.155.2.3 ]
[   FRAME #1  ]
[   SENDING   ]
```

Flow Diagram

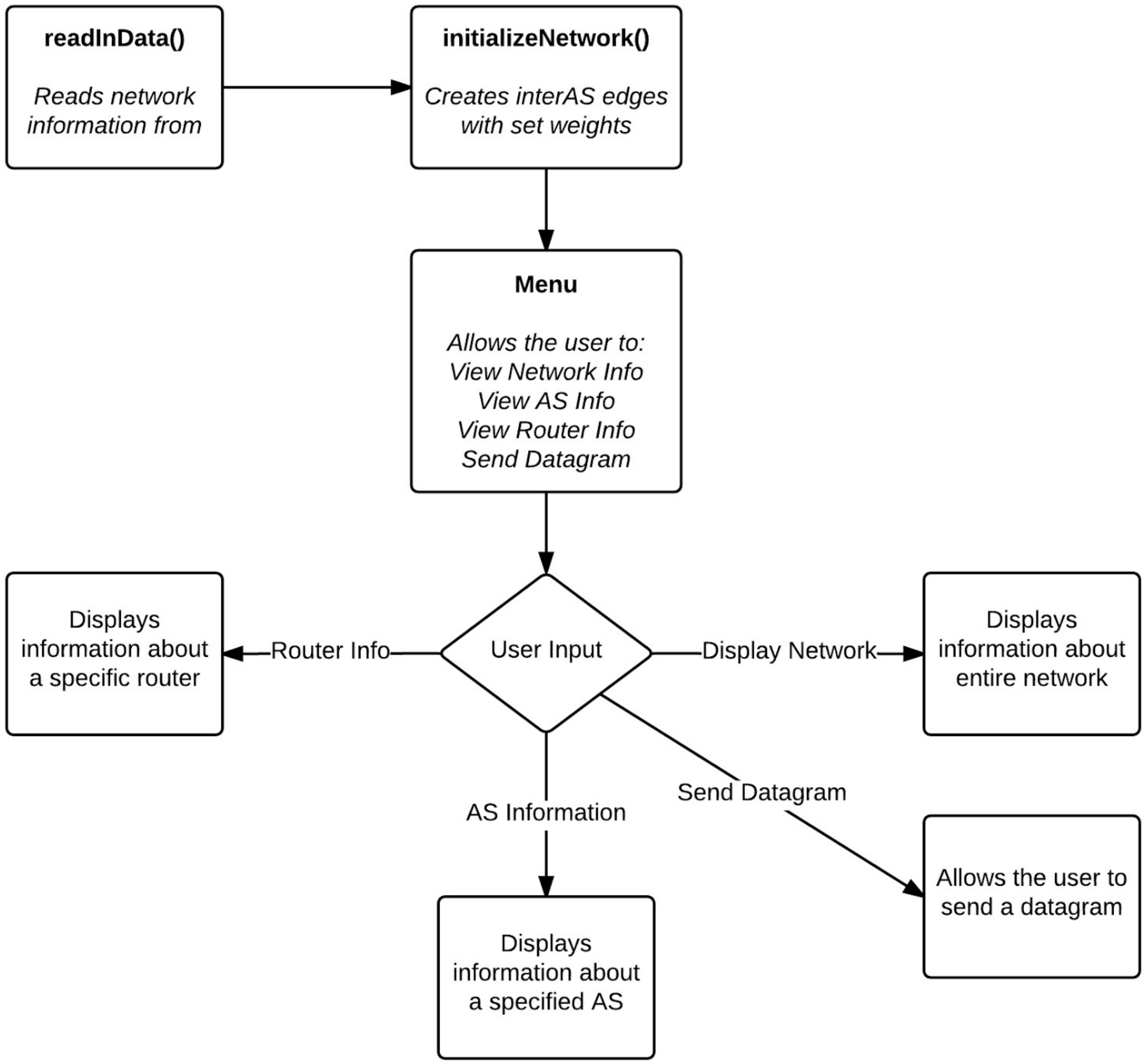


Fig. 1
Autonomous system default setup, emphasizing details of AS1

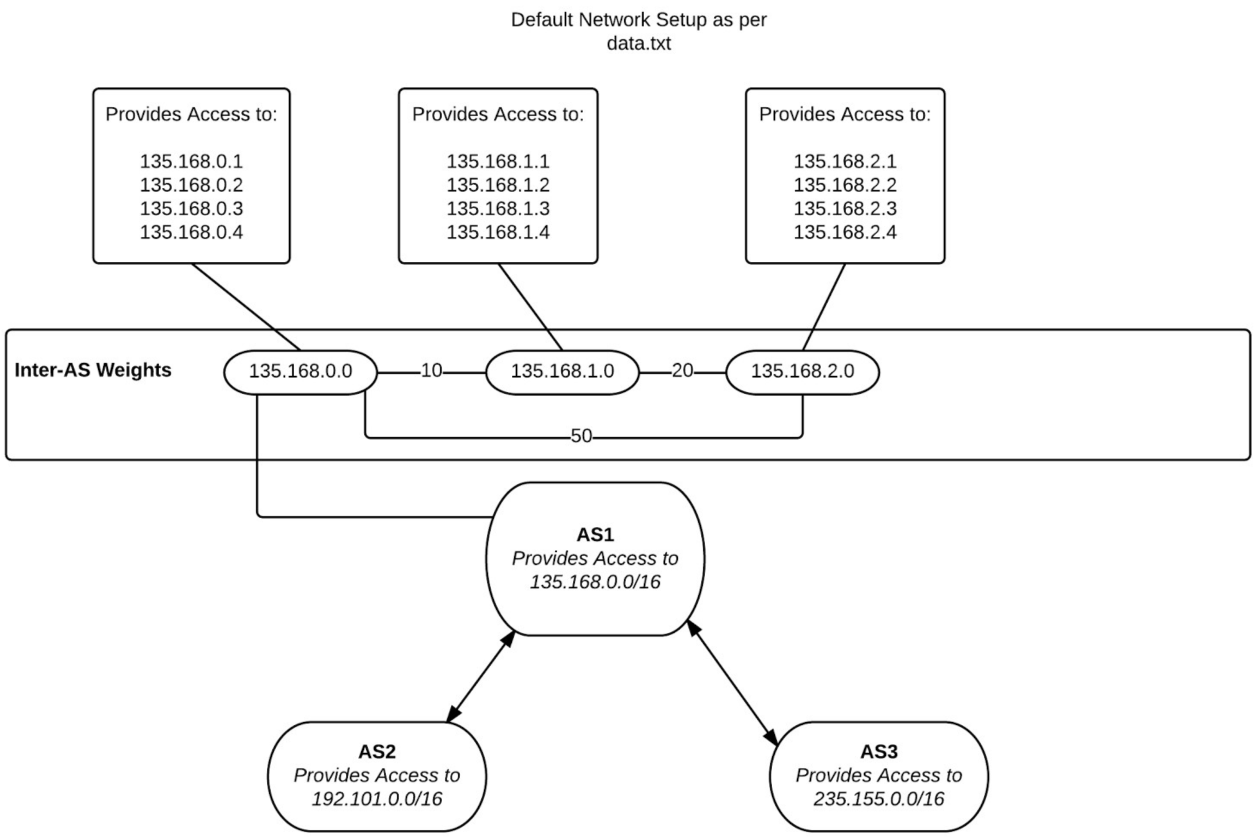


Fig. 2

This diagram shows the default setup of AS1, AS1 provides access to the 135.168.0.0/16 subnet, all traffic will then go through 135.168.0.0. Inter-AS weights represent the cost between each internal router, since only one internal router provides access to the AS.