Project: Internet paths

João Luís Sobrinho

November 3, 2020

1 Commercial paths and BGP

This project is set on the same network context as the first project. We consider internets where every pair of nodes has either a provider-customer relationship or a peer-peer relationship. A commercial path in an internet was defined as a path of the form PRC, where: (1) P is either the empty path or a path comprised solely of customer-to-provider links; (2) R is either the empty path or a single peer-to-peer link; (3) C is either the empty path or a path comprised solely of provider-to-customer links. We now define a *customer* (*peer*, *provider*) *path* as a commercial path whose first link is a customer (peer, provider) path. A customer path is preferred to a peer path, which is preferred to a provider path.

The ASes that comprise the Internet elect paths with the Border Gateway Protocol (BGP). This protocol instantiates a separate computation process per destination. For a given destination, a node learns paths from each of its out-neighbors, elects a preferred path among the learned paths, and exports the elected path to each of its in-neighbors if, and only if, the resulting path at the in-neighbor is a commercial path. A *stable state* of BGP is a state without routing messages in transit across the links of the network. We are interested only on the stable states of BGP.

In the internet of Figure 1, a provider is joined to a customer by a solid line, with the provider drawn higher than the customer, and two peers are joined by a dashed line. Taking node u_6 as destination, we have that: (1) nodes u_8, u_9, u_{10}, and u_{11} elect a customer path to reach u_6; (2) node u_7 elects a peer path to reach u_6; and (3) nodes u_1, u_2, u_5, u_3, and u_4 elect a provider path to reach u_6.

An interesting problem is to compute the *type* of path — *customer*, *peer*, *provider*, or *invalid* — elected by BGP from a source node to a destination node in an internet.

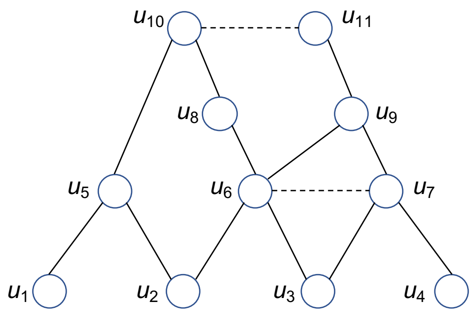

Figure 1: An internet. Solid lines join providers to customers, with providers drawn higher than customers. Dashed lines join peers.

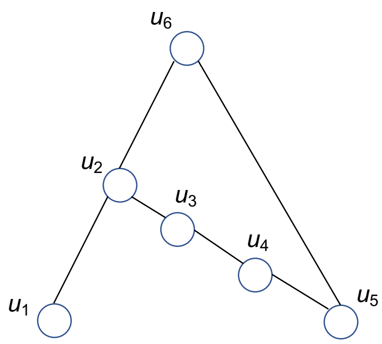

Figure 2: An internet showing that the length of the paths computed by BGP is not necessarily the shortest possible among paths of the same type.

2 Path lengths

The election of a path performed by BGP is slightly more complex than described above. Every node tie breaks among paths of the same type using their lengths, measured in terms of links. For example, in Figure 1, node u_6 elects a peer path of length 2 to reach u_4 and node u_5 elects a provider path of length 5 to reach node u_4. Therefore, node u_2 learns from u_6 a provider path of length 3 to reach u_4 and it learns from u_5 a provider path of length 6 to reach u_4. Since among paths of the same type the preference is for shorter paths, node u_2 elects the provider path of length 4 to reach u_4.

An interesting problem is to compute the length of the path elected by BGP from a source node to a destination node in an internet.

It turns out that the length of a path computed by BGP is not necessarily the length of the shortest possible path of the same type to reach the destination. This is shown in Figure 2. Node u_2 elects customer path $u_2u_3u_4u_5$ to reach node u_5, because the alternative path $u_2u_6u_5$ is a provider path. Thus, node u_1 elects provider path $u_1u_2u_3u_4u_5$ of length 4 to reach u_5. But, note that the shortest provider path from u_1 to u_5 is path $u_1u_2u_6u_5$ of length 3.

An interesting problem is to compute the length of the shortest path of the same type as that computed by BGP from a source node to a destination node in an internet.

3 Your assignment

What you have to do. You will be given an internet in the same format as that of the first project. You are asked to design and implement algorithms to solve the three problems enunciated above and produce statistics for the internet `LargeNetwork.txt` that was provided for the first project. Specifically, you are asked to:

- Design and implement a sequential algorithm that receives as input an internet, a source and a destination in that internet, and produces the type of path elected by BGP from the source to the destination. Produce the probability density function of the type of path elected by BGP over all source-destination pairs in the internet.

- Design and implement a sequential algorithm that receives as input an internet, a source and a destination in that internet, and produces the length of the path elected by BGP from the source to the destination. Produce the complementay cumulative distribution function of the lengths of the paths elected by BGP over all source-destination pairs in the internet.

- Design and implement a sequential algorithm that receives as input an internet and a destination in that internet, and produces the length of the shortest path of the same type as that computed by BGP from the source to the destination. Produce the complementay cumulative distribution function of the length of the shortest path of the same type as that computed by BGP over all source-destination pairs in an internet.

What you have to deliver, how, and when.

- You have to deliver your code and a report with a cover page and no more than three other pages containing a clear explanation of your algorithms, the statistics asked for the internet provided, and a short discussion.

- The code and the report should be sent in a .zip file to my email address with subject p2.<group number>.zip where <group number> is your group number.

- The deadline is November 29, 2020, 23:59.

How I will evaluate your assignment.

- I will start by reading your report, which is the means for you to communicate your ideas, algorithms, and conclusions with others. Organize your report in sections; present high-level, but precise descriptions of your algorithms, highlighting their most subtle steps, if any; draw concise, but unambiguous conclusions.
- If the report is readable, I will run some tests on your programs.
- I will have a discussion with you about your report and will test your code at the end of the semestre, jointly with the other assignment. All students of the group must actively participate in the design and implementation of the algorithms.