Second mini-project: Inter-domain routing

João Luís Sobrinho

I. Introduction

The Internet is an interconnected collection of networks, also called Autonomous Systems (ASs) or domains. Every pair of neighboring ASs establishes a commercial agreement between them. With some simplification, these commercial relationships belong to one of two types. In a *provider-customer* relationship the customer pays the provider to transit its traffic to and from the rest of the Internet, whereas in a *peer-peer* relationship the two peers exchange traffic between themselves and their customers without monetary compensations. We assume that there are no provider-customer cycles, where each node is a customer of the next around the cycle. Therefore, the provider-customer relationships determine an hierarchy among the nodes, with a provider at higher level of the hierarchy than its customers. At the top of the hierarchy are nodes without providers. They are called Tier-1. There are a dozen of them, including Level3, TeliaSonera, AT&T, Verizon, Deutsche Telekom, Telefonica, and NTT. At the bottom of the hierarchy are nodes without customers. There are tens of thousands of them. Figure 1 shows a small internet, where a provider is joined to a customer by a solid line and two peers are joined by a dashed line. For example, u_6 is a provider of both u_2 and u_3; u_6 and u_7 are peers. Nodes u_{10} and u_{11} are Tier-1 and nodes u_1, u_2, u_3, and u_4 are stubs.

II. Routing according to commercial policies

The commercial relationships between ASs determine the routing policies configured by the network operator of each AS. We assume that operators follow the so-called Gao-Rexford routing policies [1].

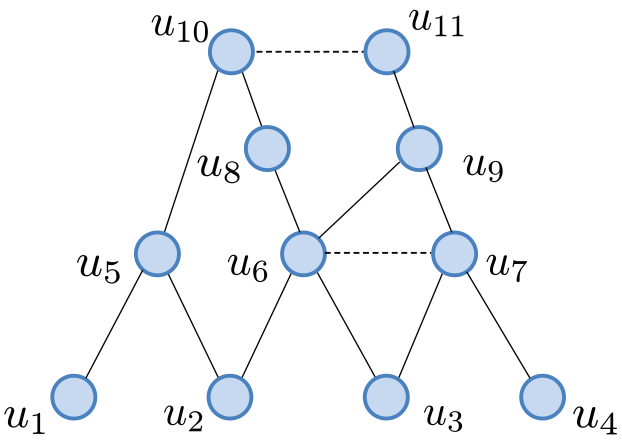

Fig. 1. An internet. Solid lines join providers to customers, with providers placed higher than customers. Dashed lines join peers.

- An AS prefers a *customer route* (learned from a customer) to a *peer route* (learned from a peer), and prefers a peer route to a *provider route* (learned from a provider). The AS elects the most preferred of the routes learned from its neighbors.
- An AS exports all elected routes to its customers and exports customers routes to all its neighbors, these being the only exportations allowed.

A *customer path* is a path P all links of which join a provider to a customer. A *peer path* is a path of the form uvP where u and v are peers and P is a customer path. A *provider path* is a path of the form PQ where P is a non-trivial path all links of which join a customer to a provider and Q is either a customer or a peer path. Customer, peer, and provider paths are *usable*; all other paths are unusable. In Figure 1, $u_{10}u_8u_6u_3$ is a customer path, $u_{10}u_{11}u_9u_7u_4$ is a peer path, $u_1u_5u_{10}u_8u_6u_3$ is a provider path, and $u_5u_2u_6$ is unusable. It is easy to show the following correspondences between elected routes and paths.

- If u elects a customer route to reach t, then there is a customer path from u to t.
- If u elects a peer route to reach t, then there is not a customer path from u to t, but there is a peer path from u to t.
- If u elects a provider route to reach t, then there is neither a customer nor a peer path from u to t, but there is a provider path from u to t.
- If u does not elect any route to reach t, then all paths from u to t are unusable.

In Figure 1, u_7, u_9, and u_{11} elect a customer route to reach u_4; u_6 and u_{10} elect a peer route to reach u_4; u_1, u_2, u_3, u_5, and u_8 elect a provider route to reach u_4.

III. Routing according to commercial policies and number of hops

At an extra level of detail, we realize that routes exchanged between neighbor nodes carry the number of hops from where they were originated up to the node currently holding the route. Among customer routes, among peer routes, or among provider routes, a node elects the route with the minimum number of hops. For example, in Figure 1, node u_9 elects a customer route with two hops to reach u_4, corresponding to customer path $u_9u_7u_4$. Node u_3 learns from u_7 a provider route with two hops, corresponding to provider path $u_3u_7u_4$, and from u_6 a provider route with three hops, corresponding to path $u_3u_6u_7u_4$. Thus, node u_3 elects the provider route with two hops to reach u_4. Node u_2 learns from u_6 a provider route with three hops, corresponding to path $u_2u_6u_7u_4$, and from u_5 a provider route with six hops, corresponding to path $u_6u_5u_{10}u_{11}u_9u_7u_4$. Therefore, node u_2 elects the provider route with three hops to reach u_4.

IV. Your assignment

What you have to do.

- Design and implement an algorithm that, given a network topology and a destination in that topology, computes the type of commercial route (provider, peer, customer, or unusable) elected at every node to reach the destination. The network topology is given as a file in the format shown below.

4323	12122	1
12122	4323	3
7018	17228	1
17228	7018	3
29017	34309	2
34309	29017	2

Each line represents a link. The first value is the identifier of the node at the tail of the link. The second value is the identifier of the AS at the head of the link. The third value is 1 if the tail of the link is a provider of the head of the link; it is 2 if the tail of the link is a peer of the head of the link; and it is 3 is the tail of the link is customer of the head of the link. For example, in the table above, 4323 is a provider of 12122. (50% of the grade.)

- Design and implement an algorithm that, given a network topology and a destination in that topology, computes the type of commercial route (provider, peer, customer, or unusable) *and the number of hops* in the route elected at every node to reach the destination. The network topology is given as a file in the same format as before. (25% of the grade.)
- Using the algorithms developed above, compare the number of hops in elected routes with the number of hops in shortest paths. The comparison can be based on the cumulative complementary distribution functions over all pairs consisting of an origin and a destination. (25% of the grade.)

What do 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 text explanation of your algorithms, their pseudo-codes, their asymptotic complexities, the statistics required, 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 11, 2016, 23:59.

How I will evaluate your assignment.

- Write your report and your pseudo-code clearly, and present a commented code.
- Be sure to test your code for correctness. I will take into consideration the efficiency of your algorithms.
- 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 assignments.

References

[1] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 681–692, December 2001.