UNIVERSITÀ DEGLI STUDI DI PISA

Dipartimento di Ingegneria Dell'Informazione

Corso di Laurea Magistrale in Computer Engineering

Advanced Network Architectures and Wireless Systems' Project

MRT DATA ANALYSIS AND ELABORATION IN ORDER TO PROVIDE A REAL-TIME AS-LEVEL TOPOLOGY GRAPH VISUALIZER

Students:
PIETRO DE ROSA
LEONARDO URBANO

A.Y. 2015/2016

Contents

Chapter 1

Background

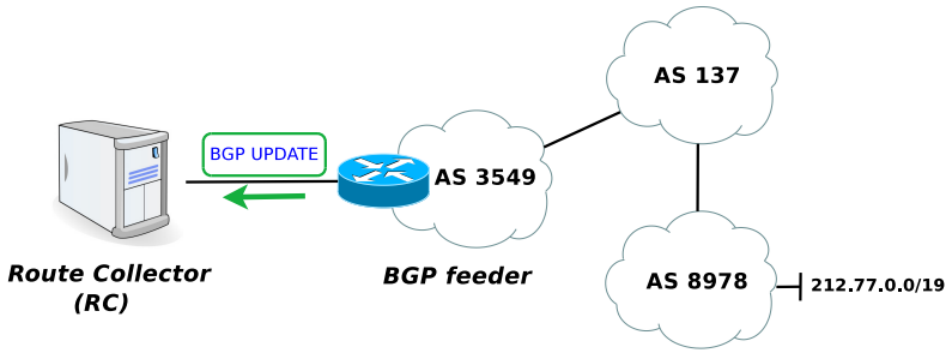
Why is it important to reveal the Internet structure? For example,

- To understand how packets are routed in the Internet and so plan an optimal inter-domain network configuration to maintain an acceptable level of service in case of malicious or unintentional faults.
- To create models of the global Internet growth and so build more realistic topology generators to simulate the Internet.

A way to know the internet topology (at AS-level) is to analyze the BGP data published by *Route Collectors*. Indeed, the Autonomous System (AS) level topology allows you to represent how the organizations, that constitute the Internet, are connected. In this topology, the Internet is modeled as a graph in which each node represents an AS and each edge represents at least one inter-domain connection established between pairs of ASes via the Border Gateway Protocol (BGP).

1.1 Route Collecting

In the late '90 the Internet community started to setup *BGP Route Collector Projects* with the aim to help the troubleshooting of reachability issues. More in detail, a *Route Collector* (RC) is a device that mimics a BGP router, with the purpose of gathering *BGP update messages* from cooperating ASes. Collected messages contain announcements of new routes and/or withdrawals of previously announced routes and are used by the RC to update its Routing Information Base (RIB) table, like a real AS Border Router (ASBR). These projects publicly release, typically in *MRT format*, periodic snapshots of the RIB of each RC and dump the related BGP update messages.



Route Collector

The Route Collector Projects we have considered are *RIS*, *Route Views* and *Isolario*. Ris and Isolario upload data every 5 minutes, while Route Views every 15 minutes. The data are collected in *MRT format*, which was developed to encapsulate, export and archive routing information in a standardized data representation .

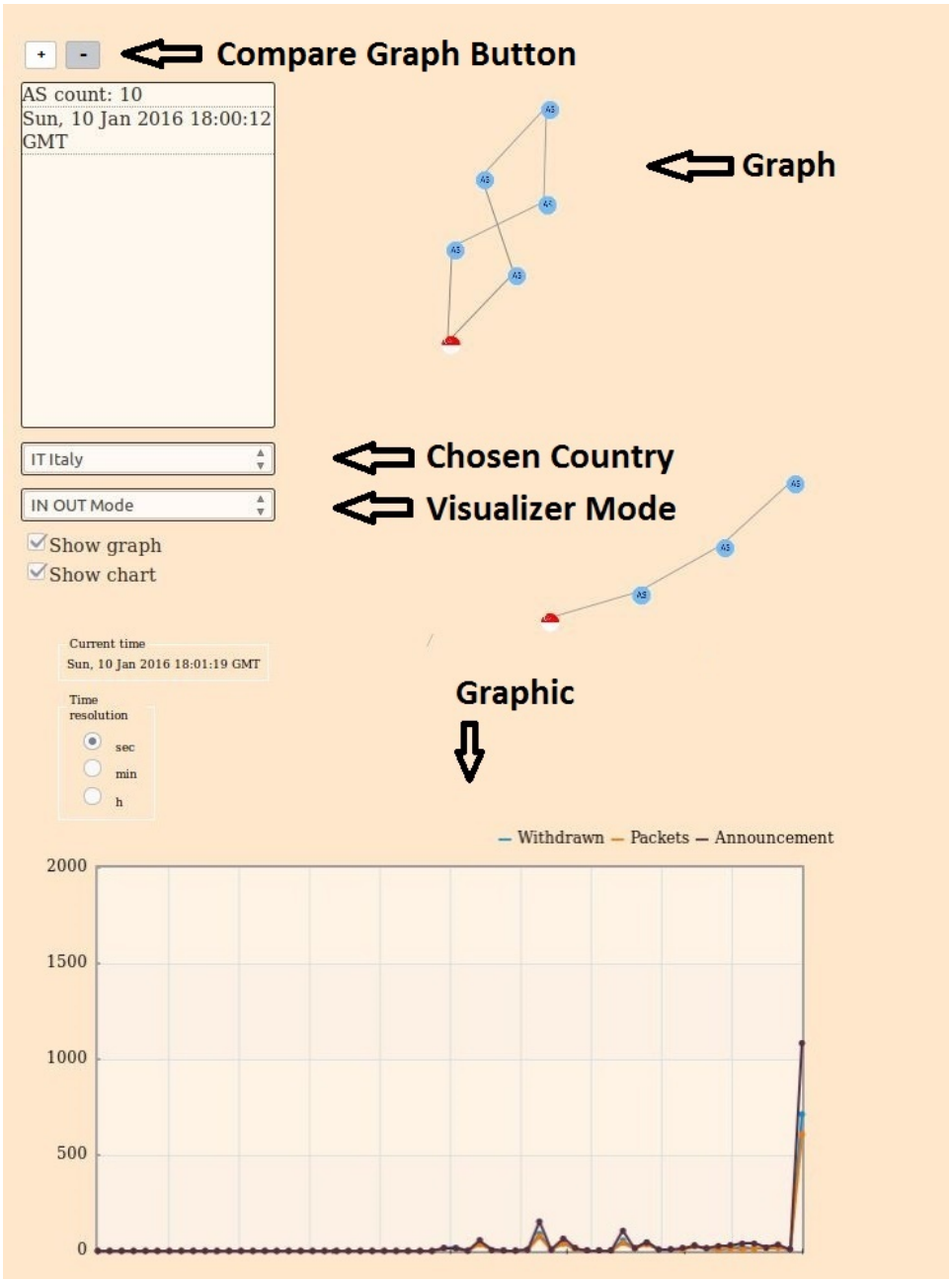
Chapter 2

Objectives

2.1 Overview

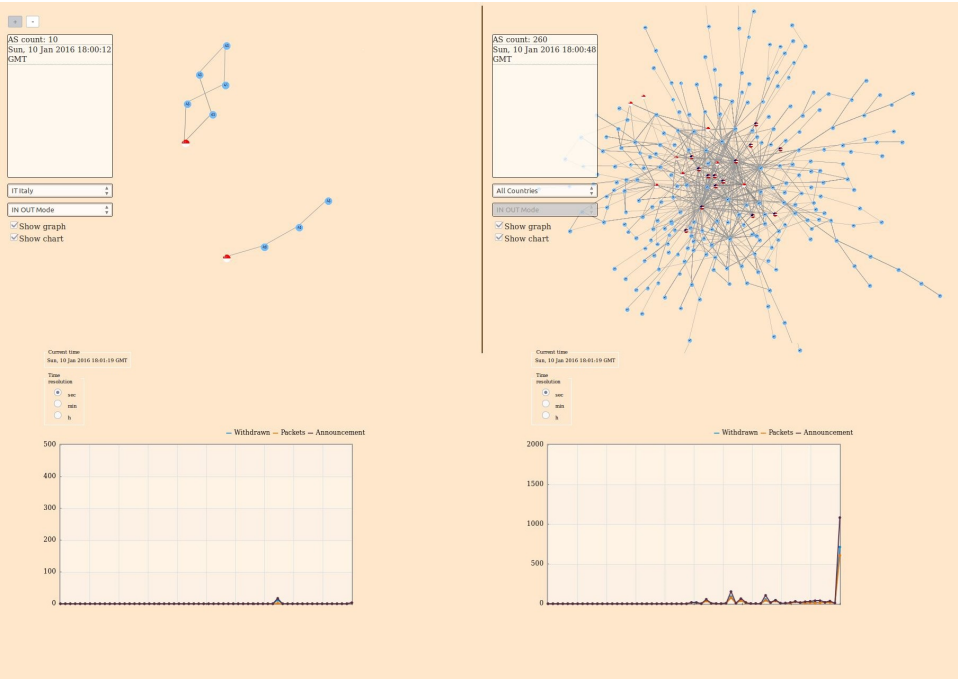
The goal of this project is to create a new service based on data collected by various route collecting projects, in order to provide a (pseudo) real-time AS-level topology graph visualizer.

We created a new service, called *Topology visualizer*, to show either the structure of the Internet AS-level topology in (pseudo) real-time, either a *chart*, related to the *graph*, that counts the number of packets, withdrawn routes and announced routes arrived in the last second, minute or hour. All of this information depend on the filters imposed by the users. In particular, the user can analyzes any *country* he/she desires and infers useful graph theory information. Furthermore, the user can choose the *visualizer mode*. There are three modes: the *Out-Mode* allows him to visualize the graph of the routes announced by the ASes of the chosen country, towards the external. The *In-Mode* allows him to visualize the graph of the routes of the chosen country, announced by the external. The *In-Out-Mode* allows him to visualize the graph of the routes announced from or towards the chosen country.



Topology Visualizer

Another functionality of the Topology Visualizer is that you can compare the evolution of the AS-level graph related to a country with the evolution of the graph of an other country. To this end, you just click the *Compare Graph Button*.

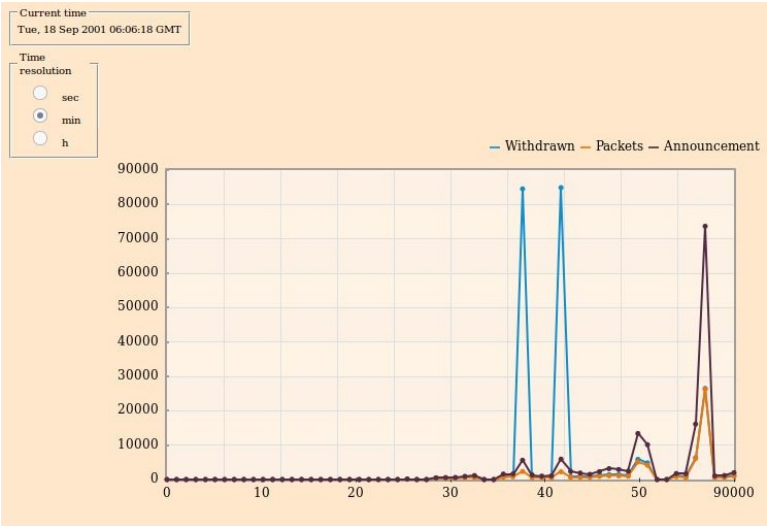


Topology Visualizer with Compare Graph activated

2.2 Extra

The analysis of the BGP data allows you also to discover Geopolitical events or worm attacks which could shutdown many subnets, and so produce a lot of traffic of BGP packets exchanged between routers.

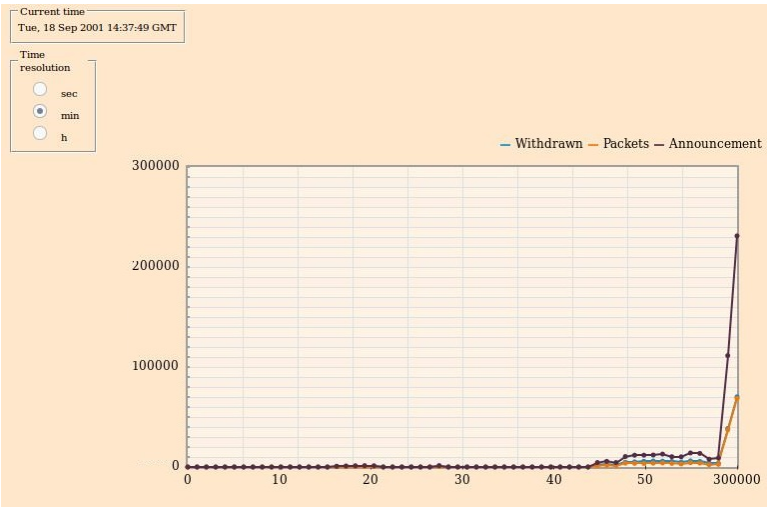
With our *Topology Visualizer* it is possible to see the general effects of this events on the AS-level topology. To this end, we have analyzed the *Nimda worm attack* occurred on 18 September 2001 that hits many American subnets. As you can see in the picture below, there is a peak of Withdrawn Routes around 5:40AM, that is the starting time of the attack.



Nimda worm attack

As you can see in the picture below, there is a subsequent (after some minutes) peak of announcement routes. It's due to the activation of backup links.

In this other picture you can see the moment in which the subnets affected have been recovered at around 2:00PM and so there is a peak of announcement routes.



Peak Announcement Routes

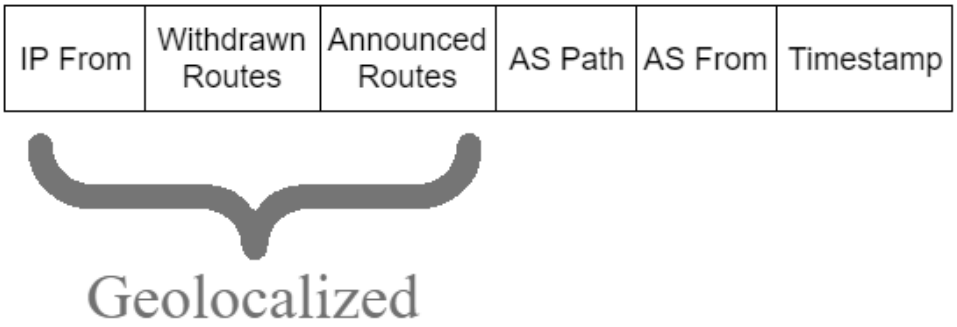
Our Topology Visualizer permits only a general analysis of worm attacks effects. For a better analysis, for example, it could show to user the details of countries and subnets involved.

Chapter 3

Implementation

The BGP protocol (nay BGP Update Messages) allows us to construct an AS-level graph of the Internet. The AS-level topology is a good abstraction of the Internet topology.

The graph is displayed at the client side but the stream that contains BGP info, useful to construct the graph, is created at the server side by the *MRT Engine*; the latter downloads and processes MRT data and, finally, creates the stream for the clients. The stream is composed by *messages* that have the following format.



Message Format

More in detail, each message is related to a BGP Message Update, indeed, the fields of a message are fields of a common BGP Update Message too. The main difference is that, in the message, some fields are *geolocalized* (IPfrom, Withdrawn and Announced routes). The Geolocalizator we've used (and provided by IIT-CNR), allows to geolocalize a subnet knowing its IP address.

3.1 Topology Visualizer

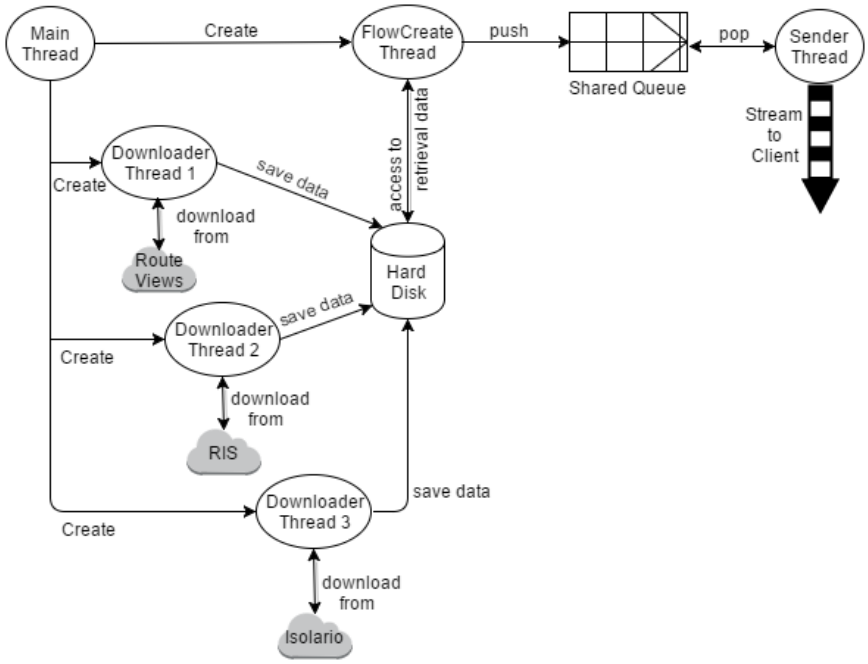
At the Client side we have developed the *Topology Visualizer* that shows the Internet's topology at AS-level (i.e. a graph where each node is an AS) and

a chart related to the graph. From an Internet AS-level analysis perspective, the most interesting path attribute, that is exchanged between BGP speakers, is the AS PATH. This attribute is mandatory and well-known, and contains the sequence of ASes that are traversed to reach the announced destinations. This attribute is originally created by the AS border router (ASBR) which owns the announced prefixes contained in the Network Layer Reachability Information (NLRI) field, and is modified whenever another ASBR propagates the route on the Internet. This BGP feature is typically exploited by research projects to infer and analyze the Internet AS-level topology. So, we have just used the information contained in the messages (AS PATH field specially), made at server side, to construct the graph. IPfrom field (Geolocalized) is used to assign a flag to each node/AS. Withdrawn and Announced routes field are used to provide to the users the list of subnets connected to an AS. Whenever a message arrives at client side, it is discarded or used in according to the filter chosen by user. To construct the graph we have used the *VivaGraph library* that allowed us a dynamic building of the graph, and the *iChartJs library* to show the relative plot.

3.2 MRT Engine

MRT Engine, at the Server side, is composed by 5 threads that work simultaneously.

All our code is in `MRT_engine\BgpUpd_Downloader_and_Sender\src\` folder.



MRT Engine

More in detail, there is a *Main Thread* (`BgpUpd_Downloader_and_Sender.cpp`) that creates 3 *Downloader Thread* (`Threads\DownloaderThreads.cpp`) and the *FlowCreate Thread* (`Threads\FlowCreateThread.cpp`). The *Sender Thread* (`Threads\SenderThread.cpp`) is created by FlowCreate Thread.

3.2.1 Downloader Threads

Each Downloader Thread continuously downloads MRT data from a Route Collecting project and saves them in the REP folder (it stands for REPositories). As soon as you start the MRT Engine each Downloader Thread downloads an initial burst of MRT Data (i.e. all the MRT Data dumped by the Route Collecting project since 60 minutes), then it enters in a infinite loop in which it downloads the last MRT Data published. After the initial burst, whenever a new file is downloaded, the corresponding downloader thread *signal*s on a condition variable. So, if the *FlowCreate Thread* was blocked, waiting for a new file to be downloaded, it will wake up and proceed. We have used the *cURL library* that allows you to download files knowing their URLs

3.2.2 FlowCreate Thread

It takes MRT Data from all Repositories (REP folder) and pushes, sorted, the BGP Update Messages (also called BGP entries) in the *Shared Queue*. The order is based on BGP entries' timestamps. Initially, the FlowCreate Thread starts the *Geolocator* that will be used by the Sender Thread. Initially, it waits until all the downloader threads finish the burst. Then, whenever a folder is found to be empty, it waits until a new file is available in the folder, i.e. it is synchronized with the *Downloader Thread* handling that specific repository. We have used the *MRT data reader library*, provided by IIT-CNR (as well as the Geolocator), to analyze MRT Data (i.e. to take BGP entries from MRT Data).

3.2.3 Sender Thread

It takes, from the shared queue, the ordered BGP entries, creates the stream for the clients and enhances the latter with geographic information. All the fields of the message can be extracted from the BGP entry. Moreover, the IPfrom, the list of withdrawn routes and the list of announced routes are sent with geographical information (i.e. this fields are geolocalized). The websocket is event-driven; so, if there aren't any clients, no action is performed. Moreover, it waits for the SharedQueue to be non-empty; the synchronization is managed by the *SharedQueue* class itself. We used the *websocketpp library*, and so the websocket protocol, to send data to the clients.