# Introducción a los Sistemas Distribuidos (75.43)
# Trabajo Práctico N°2: BGP

Esteban Carisimo, Francisco López Destain y Pablo Gotuzzo

Facultad de Ingeniería, Universidad de Buenos Aires

14 de mayo de 2019

**Resumen**

El presente trabajo práctico tiene como objetivo el análisis de la topología de Sistemas Autónomos de Intenret. Para tal finalidad, será necesario comprender argumentos teóricos de como funciona el protocolo BGP, y las limitaciones que existen para recopilar las *aristas* que forman el grafo de la red. Además, para poder lograr el objetivo planteado, se aprenderá el uso de herramientas vincualdas al análsis y la colección de datos correspondientes a la topología de Internet.

***Palabras clave***— BGP, hijacks

## 1. Introducción teórica

La estructura de la red de Internet está conformada por la interconexión de miles de redes administrativamente independintes, llamados *Sistemas Autónomos (ASes)*. La finalidad de esta red es que bajo el protocolo IP, cualquier end-host conectado a la red, pueda comunicarse con cualquier otro, sin ningún tipo de restricción. Entonces, para poder lograr tal finalidad se necesitan cumplir dos condiciones: i) Tener un grafo completamente conexo ii) Que los dispositivos de red conozcan la ubicación de cada uno de los end-hosts conectados.

La necesidad de conexidad dentro de la red expresa la necesidad de que haya al menos un camino entre cualquier par de end-hosts conectados a la red. En otras palabras, los ASes deberán estar interconectados formando una única red o grafo conexo. A nivel físico esto se expresa por medio del establecimiento de enlaces de comunicaciones (fibras ópticas, pares trenzados o conexiones inalámbricas) entre ASes. Entonces, ante la exigencia de conexidad surge la pregunta: *¿Cómo es la topología de Internet?*, es decir, *¿Cómo se contectan los ASes para poder generar un grafo completamente conexo?*.

El interés por conocer la topología de Internet es similar, aunque parcialmente, a la necesidad que existen de que todo dispositivo conozca la ubicación de cualquier destino. Internet es una red distribuida y descentralizada, por lo cual no existe un nodo central que coordine, determine o conozca su topología. Es aquí donde surge la necesidad de utilizar protocolos de ruteo, de manera tal de distribuir a través de la red, la ubicación de cada uno de los end-hosts. En particular, el protocolo de ruteo utilizado por los ASes se denomina Border Gateway Protocol (BGP). Entonces, ¿por qué entre ASes se utiliza BGP mientras que en el interior de los ASes se usan otros protocolos?. El principal motivo es que la interconexión de ASes esta regida por intereses y acuerdos comerciales, los cuales sólo pueden se expresados por el protocolo BGP. En segundo lugar, los protocolos de ruteo interno (RIPv2, OSPF) tienen resolución a nivel de routers, el cual es un nivel de granularidad irrelevamente desde ASes externos. Es decir, desde un AS como los es la UBA (AS3449), es completamente irrelevante conocer la topología de red de routers de otro AS, por ejemplo del MIT (AS3). En este ejemplo, los end-hosts de la UBA sólo están interesados por conocer que ASes se deben atravesar para arribar al MIT, y no que routers.

## 2. Detalles del protocolo BGP

BGP es un protocolo Distance Vector, por el cual los ASes intercambian información de ruteo por medio de *anuncios* constituidos por atributos. En este trabajo prácticto, estudiaremos la topología de internet, mediante el analisis de dichos atributos. Cada anuncio contiene los siguientes elementos:

- **AS-PATH**: BGP es un protocolo Distance Vector, donde la distancia esta medida en la cantidad de ASes para alcanzar el destino. Este atributo indica la cadena de ASes que se deben atravesar para alcanzar el destino.

- **NEXT HOP**: Tal como en los protocolos de ruteo interno, Este campo indica el próximo router al cual se le deben enviar los paquetes IP.
- **Prefix**: Indica el prefijo que se quiere alcanzar.
- **Comunidades**: Este es un valioso campo opcional cuyo objetivo es poder señalizar anuncios y generar ingeniería de tráfico. Su interpretación queda fuera del alcanze de la materia.

# 3. BGPstream: Motivación, intalación y ejemplos

La resolución de este trabajo práctico se hará por medio de la utilización del servicio provisto por BGPstream [1].

## 3.1. ¿Por qué necesitamos utilizar esta herramienta?

En primer lugar, por más de que seamos activos usuarios de Internet, la información correspondiente a los anuncios BGP queda sólo confinada a los routers que manejan este protocolo. Es decir, una vez que el AS recibió los anuncios BGP, volcará esta información en sus protocolos de ruteo interno, y actualizará las rutas. Finalmente, como usuarios, no podremos concoer detalladamente los AS-PATHS ni los caminos alternativos que fueron anunciados al AS al cual estamos conectados.

Ante esta limitación, el objetivo de BGPstream es poder acceder a información BGP relevante a la topología de Internet por medio de una API. Esta información BGP fue recolectada por medio de colectores públicos de BGP, los cuales brindan a sus RIBs BGP para poder observar el ecosistema de ASes. Ahora bien, si recordamos la clase de BGP, los anuncios son discrecionales por lo cual desde distintos puntos de observación, en este caso distintos colectores, la interconexión de entre ASes se verá diferente.

## 3.2. Instalación

Para la realización del trabajo se utilizará BGPstream versión 2, el cual esta disponible para descargar en `https://bgpstream.caida.org/v2-beta`. Es importante destacar que BGPstream versión 1 y su API son completamente diferentes, haciendo que el ejemplo que se muestra a continuación no pueda ser tomado como base de la resolución del TP.

## 3.3. Antes de empezar

Es importante antes de ejecutar los scripts que contengan `pybgpstream`, definir el valor de la variable de ambiente `LD_LIBRARY_PATH`, mediante el siguiente comando.

```
$ export LD_LIBRARY_PATH=/usr/local/lib
```

## 3.4. Familiarizándose con la API y los objetos

A continuación se presenta un ejemplo cuyo objetivo es analizar los caminos (AS-PATHS) que conducen a la Universidad de Buenos Aires (AS3449). Dado que Internet es dinámico, es decir la topología puede modificarse a lo largo del tiempo, el ejemplo sólo analizará los AS-PATHS presentes el 1 de Marzo de 2017 entre 00:00 y las 00:01. Para esta finalidad, se utilizará la información BGP recopoliada por el colector `route-views.saopaulo`.

Habiendo definido el escenario, a continuación se muestra un código de ejemplo.

```
'''
IMPORTANT: Load first
export LD_LIBRARY_PATH=/usr/local/lib
'''

import pybgpstream
import pprint

# Es necesario definir un lapso de tiempo en el cual se observa la RIB.
# Para que la RIB sea consistente el lapso no debe ser muy prolongado.
time_init = "2017-03-01"
```

```
time_end = "2017-03-01 00:01"

# Definino el colector desde el cual se obtiene la RIB
# La lista completa esta disponible en https://bgpstream.caida.org/data
collector = 'route-views.saopaulo'

# Defino el target AS, en este caso es la UBA (AS3449)
# Los numeros de ASes (ASN) estan disponibles en https://bgp.he.net
target_as = 3449

# Genero la consulta a la API
stream = pybgpstream.BGPStream(
    from_time=time_init,
    until_time=time_end,
    filter="type ribs and collector %s and path %s" % (
        collector,
        target_as
    )
)

# Itero en cada uno de los anuncios
for elem in stream:
    # Filtro solo los AS-PATHS que tengan como destino
    # (ultimo AS en la cadena) a la UBA
    if int(elem.fields["as-path"].split(' ')[-1]) == target_as:
        pprint.pprint(elem.fields)
```

A continuación se muestra uno de los anuncios recolectados por este colector, en donde la UBA (AS3449) es el destino.

```
{'as-path': '264268 28329 262589 262195 11058 3597 3449',
 'communities': set(['28329:2000',
                     '28329:2200',
                     '3000:3000',
                     '3000:3001',
                     '52376:1991',
                     '52376:991',
                     '64644:10100',
                     '64644:10101',
                     '64800:3333']),
 'next-hop': '187.16.221.68',
 'prefix': '157.92.42.0/24'}
```

A continuación se detallan los elementos y la resolución del código.

En la primera parte del código de ejemplo, luego de importar las bibliotecas, se encuentran las definiciones de las variables, las cuales contienen la información del escenario a estudiar.

Luego se procede a la conformación del objeto, donde se ingresan las variables que definen el escenario. Sin embargo, en la conformación del objeto `stream`, lo más importante es la función del argumento `filter`. En este caso, por medio de `path` y `collector` reducimos el número de anuncios que responderá la API.

Finalmente, se iterará por cada uno de los elementos que corresponden a la consulta generada. Aquí, cada elemento corresponde a un anuncio BGP camino, y por ende a un camino. Sin embargo, debido a la dinámica normal del protocolo BGP, puede haber elementos replicados. Ahora, dado que sólo se busca analizar los AS-PATHS hacia la UBA, se deben filtrar todos los AS-PATHS que no contengan a AS3449 como origen (último elemento de la cadena).

## 3.5. Lista de colectores (collectors)

En el ejemplo anterior, se eligió utilizar el colector identificado bajo el nombre `route-views.saopaulo`. También existen otros colectores disponibles, en la siguiente página `https://bgpstream.caida.org/data` se detalla su identificador, ubicación y desde cuando se encuentran activo.

### 3.6. Más información

- `https://github.com/estcarisimo/fiuba-7543-BGPstream`
- Diapos de clase
- `https://github.com/CAIDA/bgpstream-tma-phdschool`

## 4. Preguntas a responder

A continuación se detallan las preguntas sobre las cuales se tendrá que trabajar.

### 4.1. Modalidad de trabajo

**El trabajo práctico se hará en grupos de dos integrantes.**

### 4.2. Detalles a tener en cuenta para la implementación

- Sea paciente puede tardar, la consulta a la API puede demorar a causa de la enorme cantidad de anuncios que existen en BGP.
- Se recomienda usar ventanas no mayores a 15 minutos

### 4.3. Preguntas cortas

En esta sección se mencionará el AS objetivo, cuyo ASN será presentado en la sección 4.6.

1. ¿Cuántos AS-PATHS diferentes se ven desde cada colector hacia el AS objetivo y hacia su ISP? Se considera un AS-PATH a toda la cadena de ASes. Usted debe mostrar 4 resultados (2 colectores, 2 ASes)
2. ¿Cuáles son los proveedores observados para el AS objetivo y para su ISP? Se considera proveedor al AS inmediatamente anterior en la cadena de AS-PATH. En la medida de lo posible trate de decir el ASname de los proveedores. *IMPORTANTE: Remueva el AS-PATH prepend*
3. ¿Cuántos prefijos diferentes anuncia el AS objetivo y cuantos su ISP?
4. Repita todos los puntos anteriores pero para el año 2012. En caso de haber diferencias, detállelas.

### 4.4. Estado de desagregación en la full RIB

Se deberá por medio de BGPstream[1] conseguir una tabla de ruteo BGP completa (full RIB), para luego inspeccionar cuantas entradas tiene la tabla antes y después de efectuar la *máxima agregación posible*.

IMPORTANTE: La agregacion de prefijos debe tener en cuenta los AS origen. Es decir, prefijos continuos pero originados por diferentes ASes no pueden ser agregados.

Para poder llevar a cabo el ejercicio deberá seleccionar:

1. **Un colector**: podrá ser cualquiera que crea conveniente de la siguiente lista `https://bgpstream.caida.org/data`
2. **Una fecha**: Deberá usar como día y mes, la fecha de cumpleaños de alguno de los integrantes, y como año, el valor 2015. El colector elegido deberá disponer de información en esa fecha.

### 4.5. Secuestros de prefijos

Deberá generar un código el cual sea capaz de detectar el secuestro de prefijos. Utilice información externa correspondiente a *secuestro de prefijos* para validar que su código funciona correctamente. Además deberá explicar cual es su hipótesis para argumentar que un prefijo fue realmente *"secuestrado"*.

[1] https://bgpstream.caida.org/

### 4.6. ¿Qué ASes le tocan a mi equipo?

Mediante el siguiente cálculo y tabla se indica el AS objetivo con el cual deberá trabajar cada grupo

$$clave = \left( \sum_{i=0}^{N} digito\_padron(i) \right) mod 10 \tag{1}$$

Donde $N$ es la cantidad de digitos en los padrones de ambos integrantes del grupo.

| Clave | ASN | ASnane | País |
|---|---|---|---|
| 0 | 23248 | Paducah Power Sys | US |
| 1 | 202167 | ZeniMax | DE |
| 2 | 8966 | Etisalat | AE |
| 3 | 262907 | Avanto | BR |
| 4 | 6057 | Antel | UY |
| 5 | 61510 | Coop. del Calafate | AR |
| 6 | 34795 | Banca d'Italia | IT |
| 7 | 36917 | Angola Com. | AO |
| 8 | 134522 | Geotel | BD |
| 9 | 18353 | Revera | NZ |

## Referencias


[1] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. Bgpstream: a software framework for live and historical bgp data analysis. In *Proceedings of the 2016 Internet Measurement Conference*, pages 429–444. ACM, 2016.