



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

## TEORIA DE ALGORITMOS II

Trabajo Practico Final  
Primer cuatrimestre 2022

---

Hojman de la Rosa, Joaquin Guido	102264	jhojman@fi.uba.ar
Giampieri Mutti, Leonardo	102358	lgiamperi@fi.uba.ar

---

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Análisis Inicial</b>	<b>3</b>
<b>3. Distribución de Grados</b>	<b>5</b>
<b>4. Centralidad</b>	<b>7</b>
<b>5. Homofilia</b>	<b>10</b>
<b>6. Comunidades</b>	<b>13</b>
<b>7. Motifs</b>	<b>15</b>
<b>8. Roles</b>	<b>18</b>
<b>9. Outbreaks</b>	<b>21</b>
<b>10. Propagación en una red: Aplicación de Cascadas</b>	<b>24</b>

## 1. Introducción

El presente trabajo final, correspondiente a la materia Teoría de Algoritmos II, consistirá en el análisis de la siguiente red: Autonomous systems AS-733, que modelaremos en forma de grafo.

El concepto de la red es el siguiente: El grafo de routers que componen internet se puede organizar en subgrafos denominados Sistemas Autónomos (AS). Cada AS intercambia flujos de tráfico (paquetes) con sus AS vecinos (peers). Los nodos del grafo a analizar serán los AS mientras que las aristas que los unen, será si estos sistemas autónomos se conocen”.

Los datos se recopilaron del proyecto Route Views de la Universidad de Oregón. El conjunto de datos fue tomado entre el 8 de noviembre de 1997 hasta el 2 de enero de 2000.

Es importante destacar que los nodos que aparecen en la red tienen un número por nombre, y este número pertenece a un sistema autónomo real, pudiendo utilizar la pagina de HackerTarget para determinar de que sistema autónomo se trata.

En este informe se analizaran diversos tópicos sobre el grafo mencionado: veremos sus características principales, las comunidades que lo conforman y los roles de cada nodo, buscaremos determinar quienes son los mas centrales e influyentes, como podrían formarse cascadas en el grafo, donde poner sensores para evitar ataques sobre la red, si existe algún tipo de homofilia y como están compuestos los clusters del grafo.

Se adjunta a continuación el repositorio de github que contiene la totalidad de los notebooks utilizados para el presente trabajo. El archivo as.txt representa al grafo.

Autonomous-systems-network-analysis

## 2. Análisis Inicial

Inicialmente, antes de comenzar con cualquier tipo de análisis sobre la red, debemos conocer algunas cuestiones básicas del grafo. Primero que nada queremos ver el grafo:



A continuación algunas características de la red:

- El grafo es NO dirigido.
- La cantidad de nodos es 6474.
- La cantidad de aristas es 13895.
- El diámetro es 9.
- El grado promedio de los nodos es de 4.29.

- El coeficiente de clustering promedio es 0.2522.

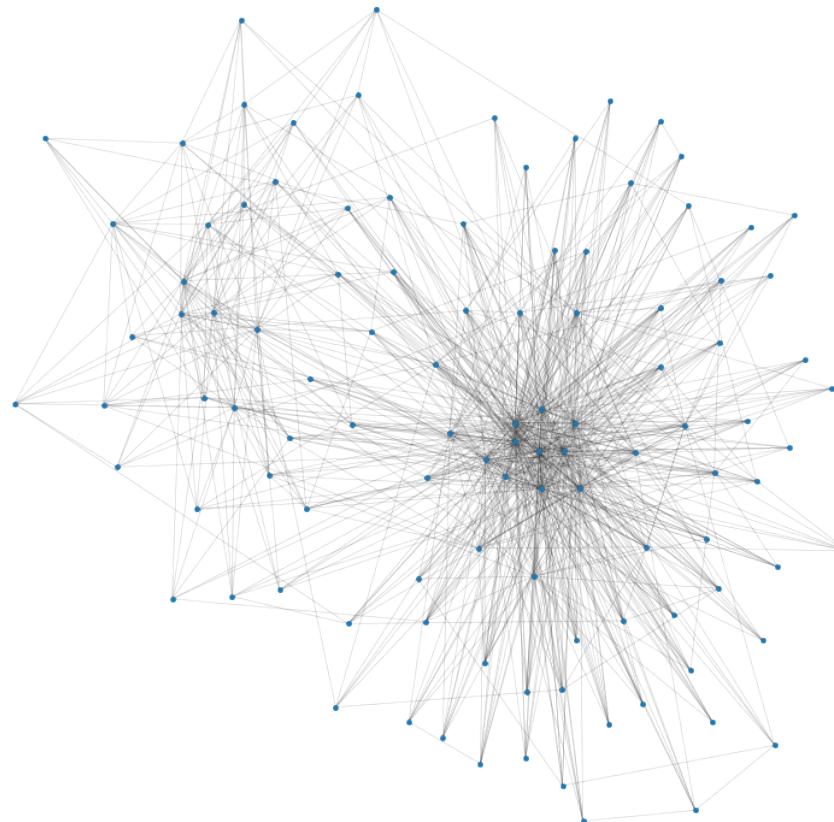
Otra característica que vale la pena comentar un poco más en detalle es que el grafo tiene 1323 self loops, son nodos que se linkean a sí mismos, lo cual en un sistema automático tiene sentido ya que un AS puede tener más de un router de borde.

Por otro lado, nos preguntamos quien es el nodo más conectado de la red. Dicho nodo resultó ser el AS1, con 1460 aristas. El nombre del AS1 es LVLT-1, es una empresa de telecomunicaciones estadounidense llamada Level 3 Communications.

El grafo está conectado, pero posee 600 puntos de articulación. Es decir que hay 600 vértices tal que si los eliminamos se produce un incremento en el número de componentes conexos, es decir que si sacamos este vértice, el grafo pasará a desconectarse. Podemos pensar que tener casi el 10 por ciento de los nodos como puntos de articulación, sumado al diámetro que es bastante elevado, nos hace estar en presencia de un grafo bastante poco completo.

Algo que refuerza lo comentado en el párrafo anterior es que en el grafo existen 2451 puentes globales, es decir aproximadamente el 17 por ciento de las aristas generaría un aumento en las componentes conexas del grafo si las eliminamos. La cantidad de puentes locales es de 2420.

Para finalizar esta sección nos pareció interesante mostrar el grafo luego de aplicar el algoritmo de k cores sobre el mismo. Un grafo de k cores es el máximo subgrafo que contiene nodos de grafo K o mayor, y en nuestro caso este valor de k es igual a 7:

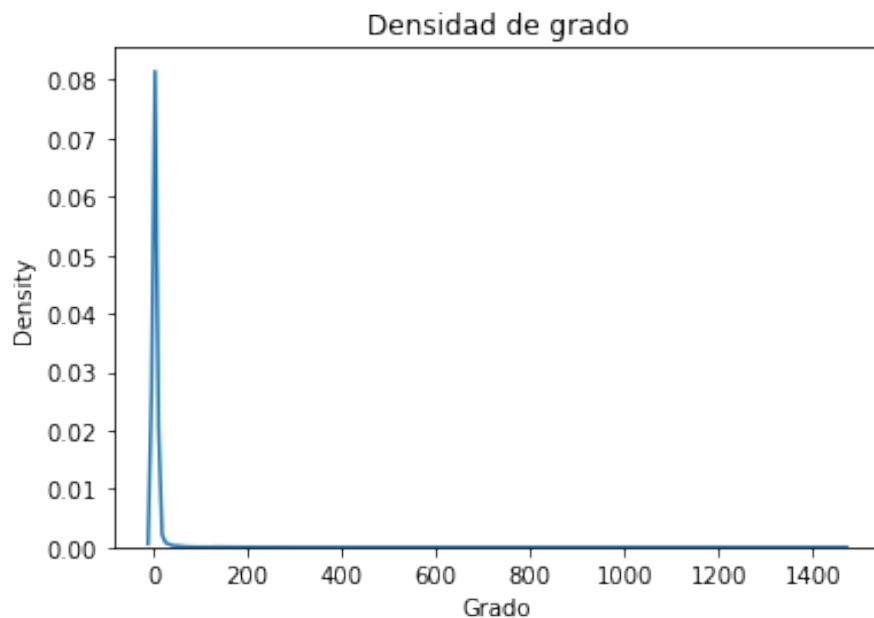


Para lograr esta visualización, debido a las características del algoritmo, se eliminaron los self loops.

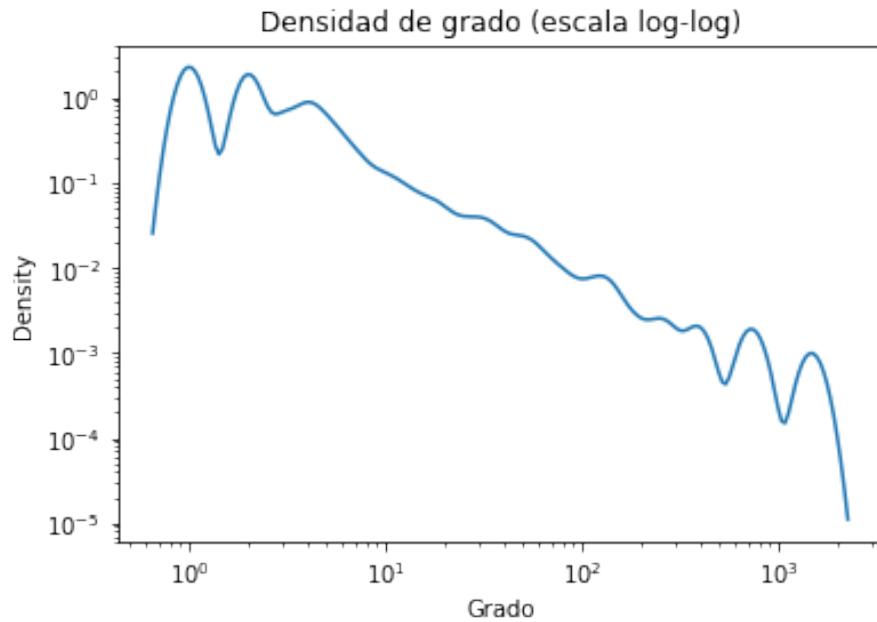
### 3. Distribución de Grados

Hasta ahora, vimos que la red tiene un diámetro relativamente pequeño comparado con la cantidad de sistemas autónomos de la red; a lo sumo, 9 routers de borde separan a los dos sistemas autónomos más lejanos, con lo cual, un paquete viajando entre redes de distintos sistemas autónomos en esta red deberá pasar 9 hops de borde, en el peor de los casos. Considerando que hay casi 7000 sistemas autónomos en esta red (que cabe destacar, no refleja la realidad, ya que en el mundo hay más de 100000, pero sin embargo se conservan las propiedades esperadas), se puede ver que esta distancia es relativamente corta. Esto tiene sentido; un paquete viajando por la red es encolado en múltiples routers para ser redirigido a destino. A mayor cantidad de routers atravesados, mayor es la probabilidad de que se droppee en alguno, por ende, el diseño de la red tiende a minimizar distancias entre los sistemas autónomos. Si bien dentro de un sistema autónomo hay múltiples subredes, la idea de minimizar distancias, ya sea al interior o al exterior del sistema autónomo, se mantiene.

Para definir con mayor detalle la estructura de la red, se realiza un análisis de la distribución de grados.



Lo que se puede apreciar es que la mayor parte de los sistemas autónomos tienen pocas conexiones, algo que reflejaba el grado promedio. Viendo en escala log-log:



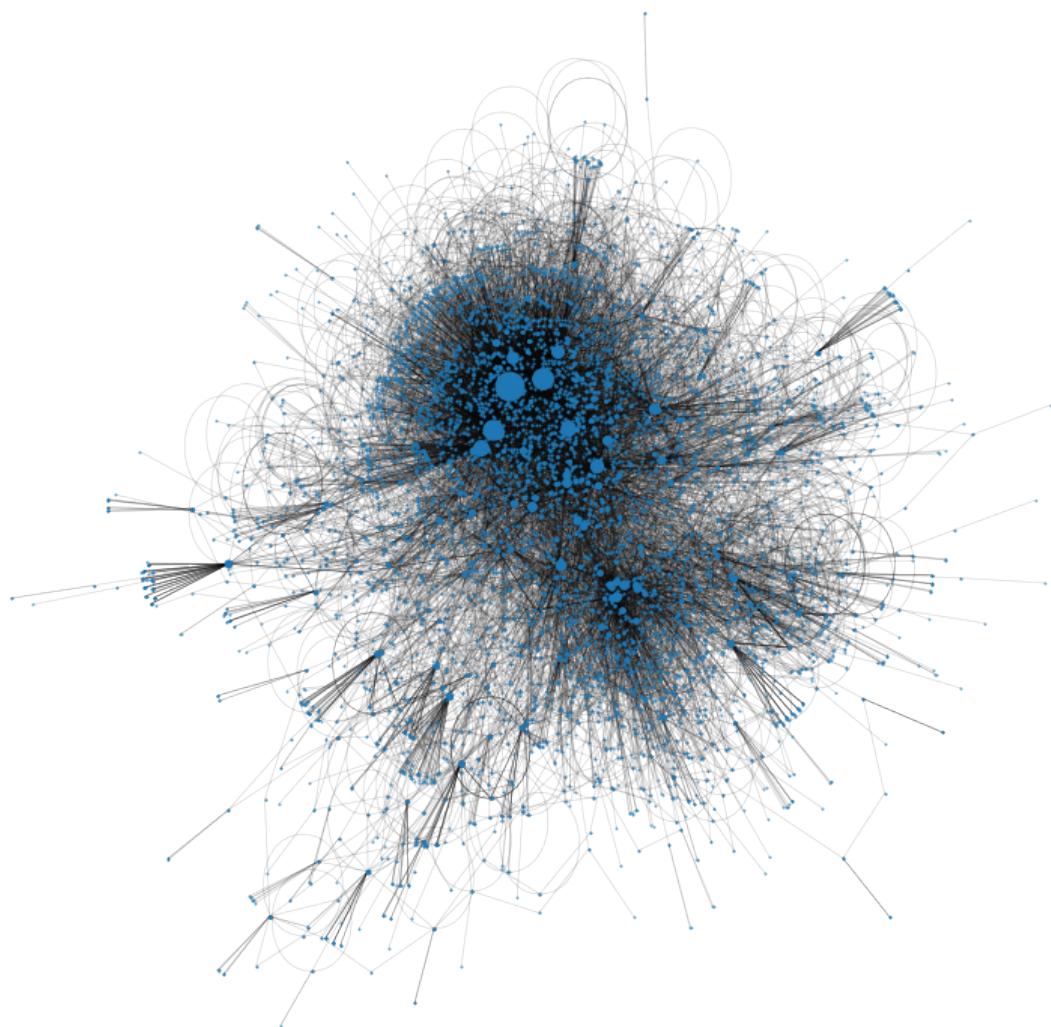
Se ve que la mayor concentración de densidad (99 % de la misma) corresponde a sistemas autónomos que tienen entre 1 y 10 vecinos, pero también se pueden apreciar picos en la densidad cuando el grado se aproxima a 1000; el sistema autónomo con más cantidad de conexiones tiene aproximadamente 1400 vecinos, que sobrepasa en 2 ordenes de magnitud a la cantidad de vecinos que tiene el 99 % de la población. Este mismo argumento se puede hacer para analizar distribuciones, como la Pareto, que siguen una powerlaw.

Mediante el estimador de máxima verosimilitud de alpha para una ley de potencias, y utilizando Kolmogorov para obtener el grado a partir del cual la distribución se comporta como una powerlaw (encontramos que a partir de 8 vecinos en adelante la densidad decae rápidamente), se obtuvo el alpha estimado  $\alpha = 2,14$ , valor que se encuentra entre 2 y 3, y es común en grafos que se comportan como redes sociales.

## 4. Centralidad

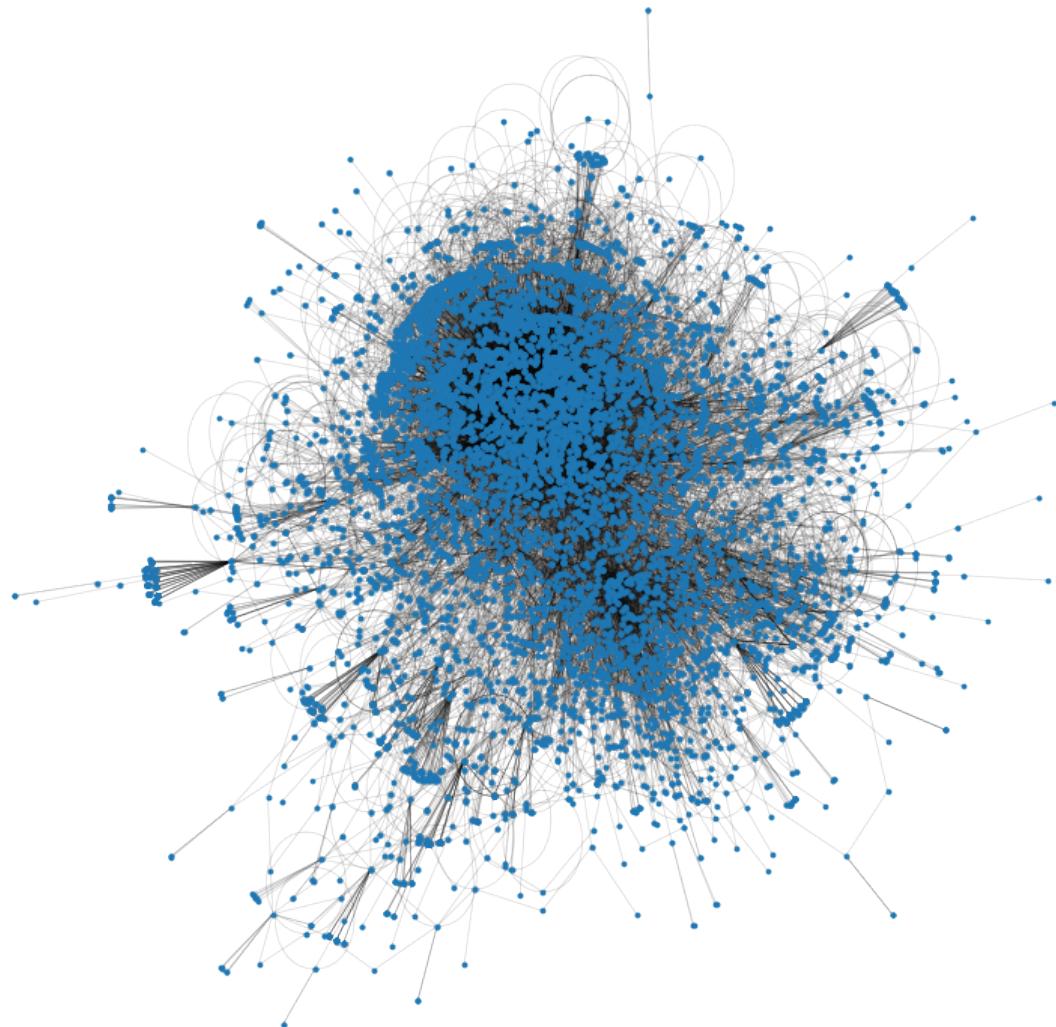
Queremos conocer la centralidad de la red. La centralidad de un grafo son aquellos vértices más importantes o influyentes, y no existe una única manera de medirla. En el presente análisis veremos varios tipos de centralidades, y luego de aplicarlos sobre el grafo, compararemos los resultados. Mostraremos el grafo en cada caso y el tamaño del nodo dependerá de su centralidad.

Aplicaremos primero una centralidad de tipo geométrica: Centralidad de Grado, donde la centralidad de un nodo depende de cuantas aristas este conectado. La centralidad de grado es relevante en la red de sistemas autónomos ya que muestra a los AS más centrales según la cantidad de conexiones. Estos serían probablemente los objetivos principales de atacantes que quieran desconectar una buena parte de Internet.

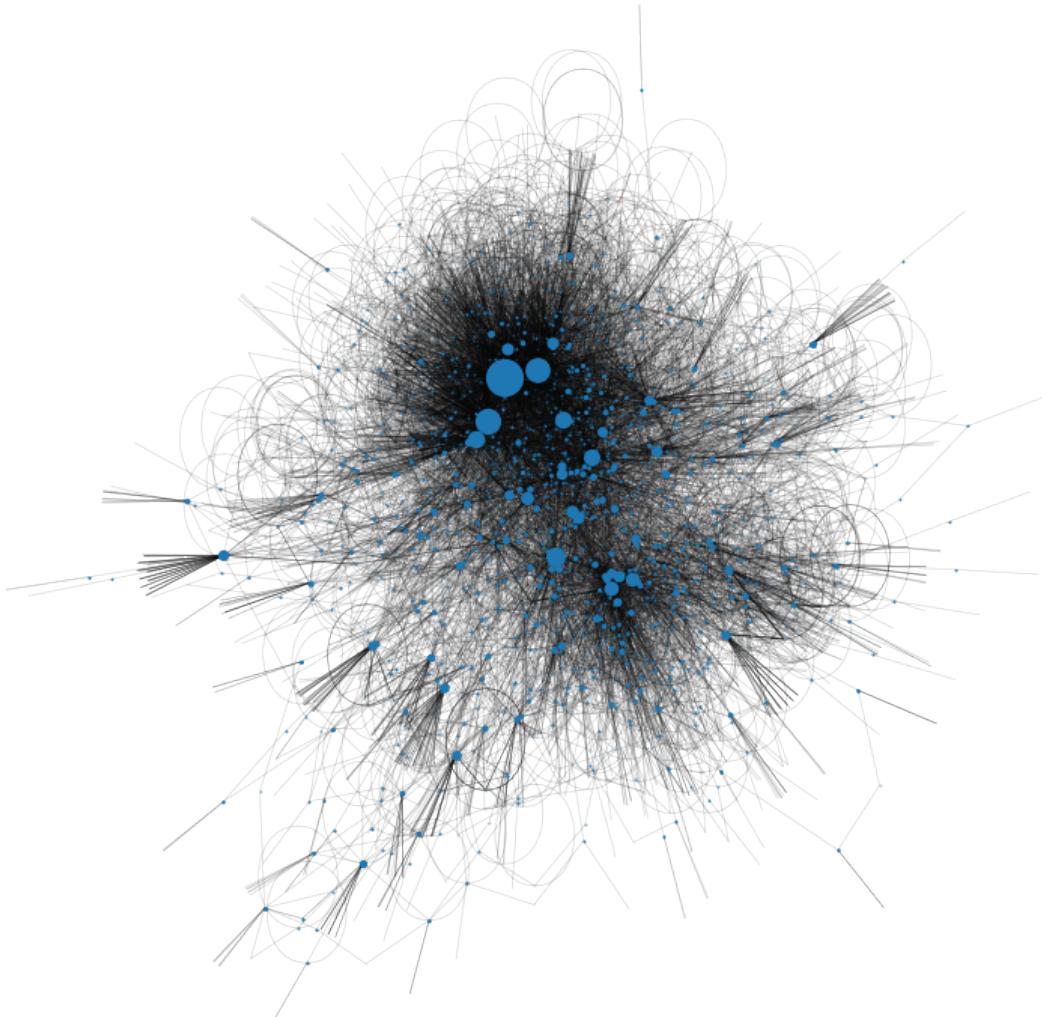


A continuación veremos otra centralidad de tipo geométrica: Centralidad de Proximidad (Closeness). Aquí los nodos más centrales van a ser los que literalmente estén en el centro. Estos son los

sistemas autónomos que tienen la mayor eficiencia y capacidad al momento de distribuir paquetes en Internet, ya sea porque estan conectados a muchos nodos o porque son vecinos de ISPs/IXPs/universidades.



Por ultimo veremos otro tipo de centralidad, esta basada en caminos: betweenness centrality. Este tipo de centralidad indica la cantidad de veces que un vértice aparece como intermediario en algún camino mínimo (donde no sea origen ni destino).



La centralidad de caminos nos muestra aquellos sistemas autónomos que funcionan como IXPs, o exchange points. Justamente, los exchange points son puntos particulares que sirven para interconectar múltiples ISPs entre sí.

Si averiguamos quienes son los nodos mas centrales en cada uno de los casos calculados vemos que... son los mismos nodos para cada centralidad! Los nodos 1, 9 y 6 son, en ese orden, los nodos mas centrales de la red.

¿Cuáles son estos Sistemas Autónomos? El 1 ya lo comentamos antes: corresponde a LVLT-1, US la cual era una empresa multinacional estadounidense de servicios de telecomunicaciones e Internet, llamada Level 3 Communications. El AS 9 corresponde al sistema autónomo llamado CMU-ROUTER, US y pertenece a la Carnegie Mellon University, y el AS6 llamado BULL-HN, US, que es de la compañía ATOS IT Solutions and Services, Inc. Todas ellas empresas estadounidenses. Estas empresas por lo general se dedicaban a funcionar como ISPs de tier 1, de ahí que conocen a una gran parte de la red.

## 5. Homofilia

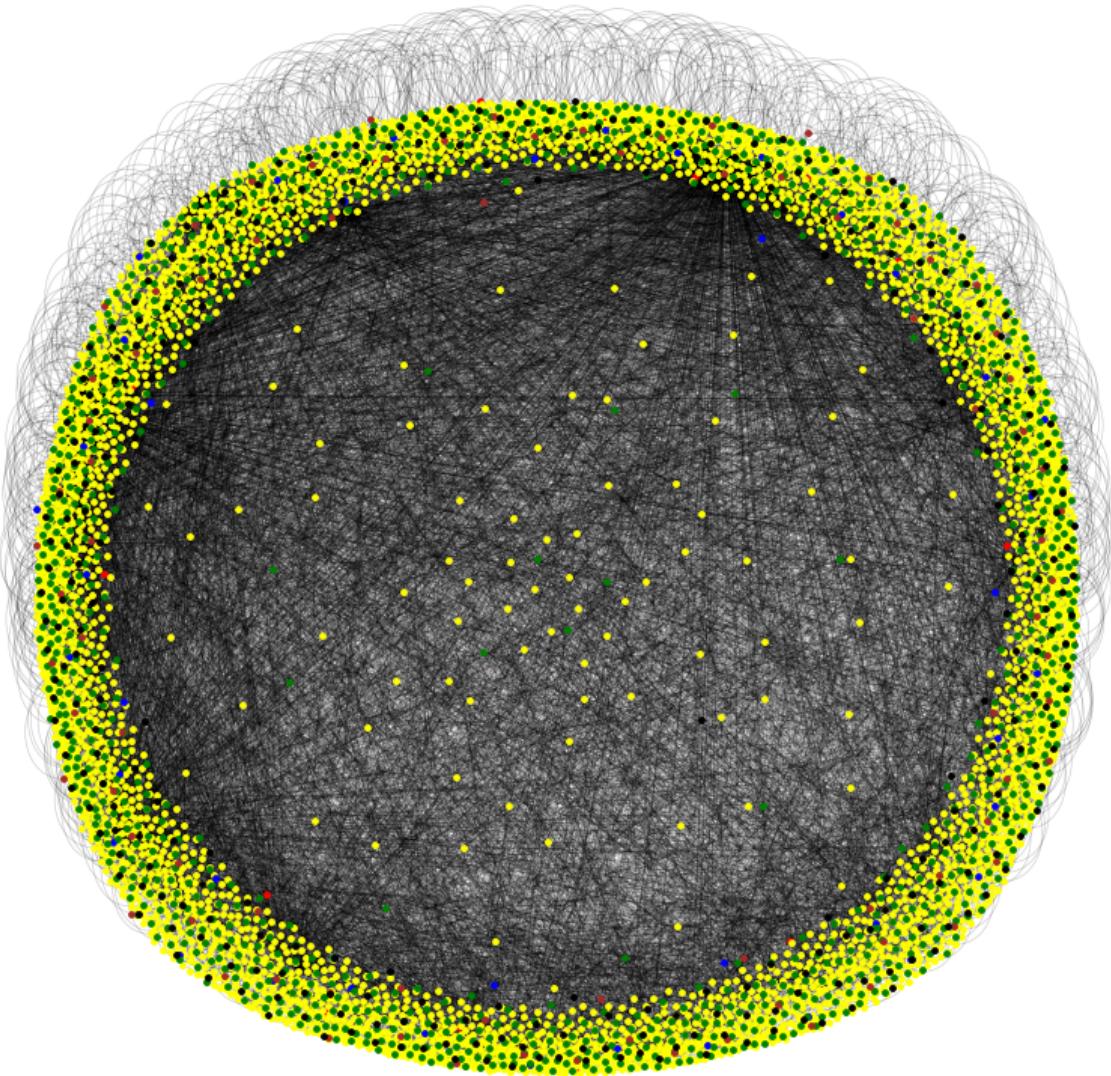
Para hacer un test de homofilia, decidimos categorizar a los sistemas autónomos por continente. La idea es ver si en la estructura de la red, los sistemas autónomos tienden a conectarse más hacia sistemas autónomos geográficamente más cercanos, es decir, a sistemas autónomos dentro del mismo continente. Para este análisis nos valimos de ciertos recursos en linea para mappear los números de los sistemas autónomos a su ubicación en el globo. Algunos de los sistemas autónomos que aparecen en nuestra red tienen números que están reservados según IANA, con lo cual no tomamos esos para el análisis. Esto último puede deberse a distintos motivos técnicos que escapan al scope del análisis y que se definen principalmente en las RFCs de BGP.

Tomando la componente conexa más grande luego de remover los nodos que no tienen un continente definido, obtenemos los siguientes valores:

Continente	Cantidad de Sistemas Autónomos
América del Norte	4686
Europa	1199
Asia	328
Oceanía	84
América del Sur	44
África	26

En total, tenemos 6367 sistemas autónomos categorizados por continente en la componente conexa más grande. La idea ahora es calcular cual es la proporción de enlaces que cruzan continente y la proporción de enlaces que, tomando una red aleatoria, deberían cruzar continente para que no haya homofilia; el threshold.

Antes, una visualización de la red categorizada.



Vemos en amarillo los sistemas autónomos de Norte América, en verde los de Europa, en negro los de Asia, en marrón los de Oceanía, en rojo los de África y en azul los de América del Sur.

La proporción de enlaces intercontinentales en la red original es de 0.324. En promedio, aproximadamente 3 de cada 10 enlaces de la red cruzan continente. Ahora, si la red y los enlaces fuesen uniformemente generados, esto es, no hubiese ningún tipo de segmentación o sesgo, la proporción de enlaces que cruzan continente debería ser de 0.420. Esto se calcula de la siguiente tabla:

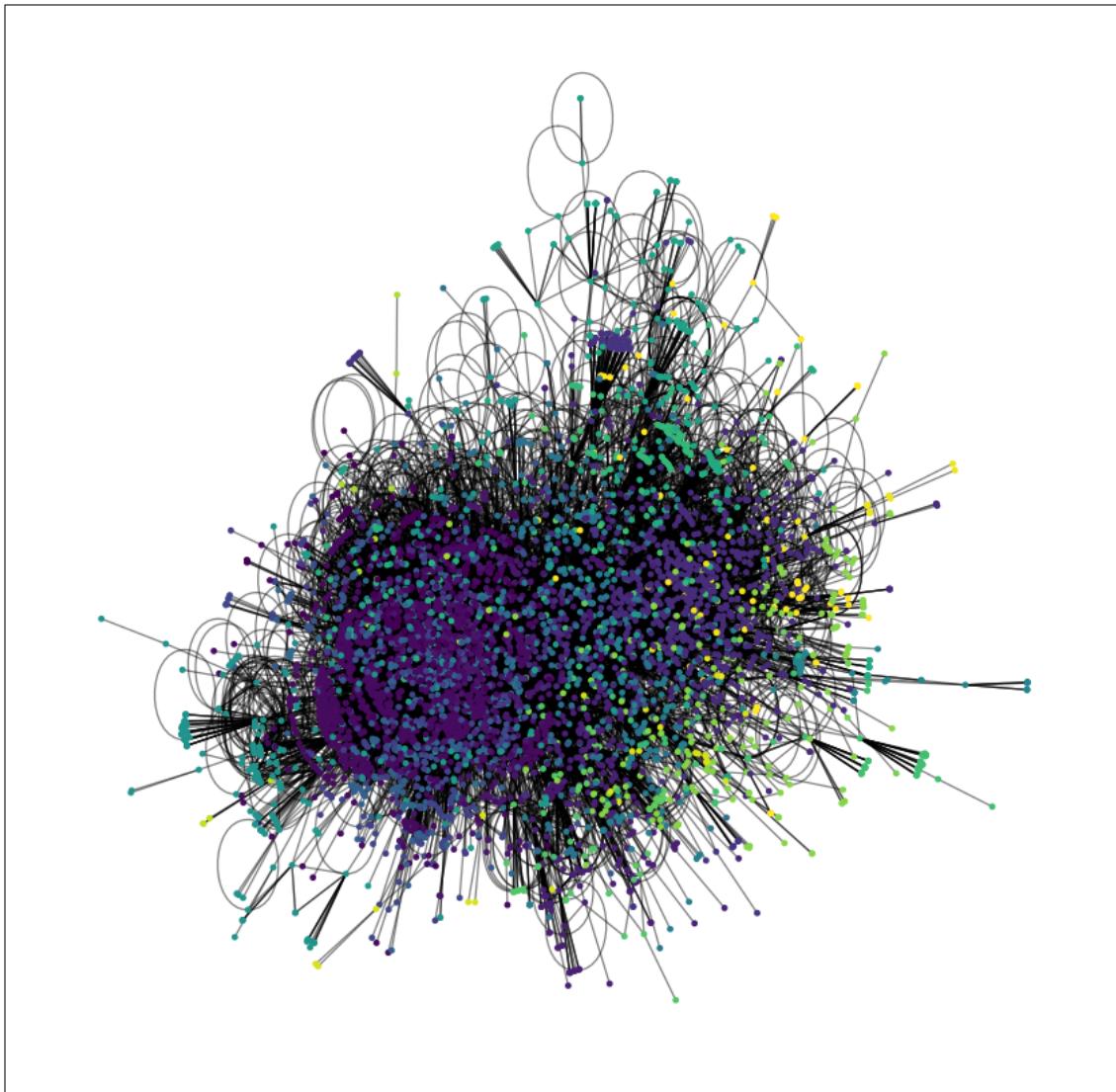
Continente	Proporción de sistemas autónomos
América del Norte	0.736
Europa	0.188
Asia	0.052
Oceanía	0.013
América del Sur	0.007
África	0.004

Por lo tanto, puede decirse que la red presenta homofilia por continente. Es algo que no sorprende debido a las conexiones físicas entre routers de borde intercontinentales son mucho más difíciles de llevar a cabo que conexiones dentro del continente, si por ejemplo pensamos en cables submarinos (una observación importante es que no todos los cables intercontinentales son submarinos).

## 6. Comunidades

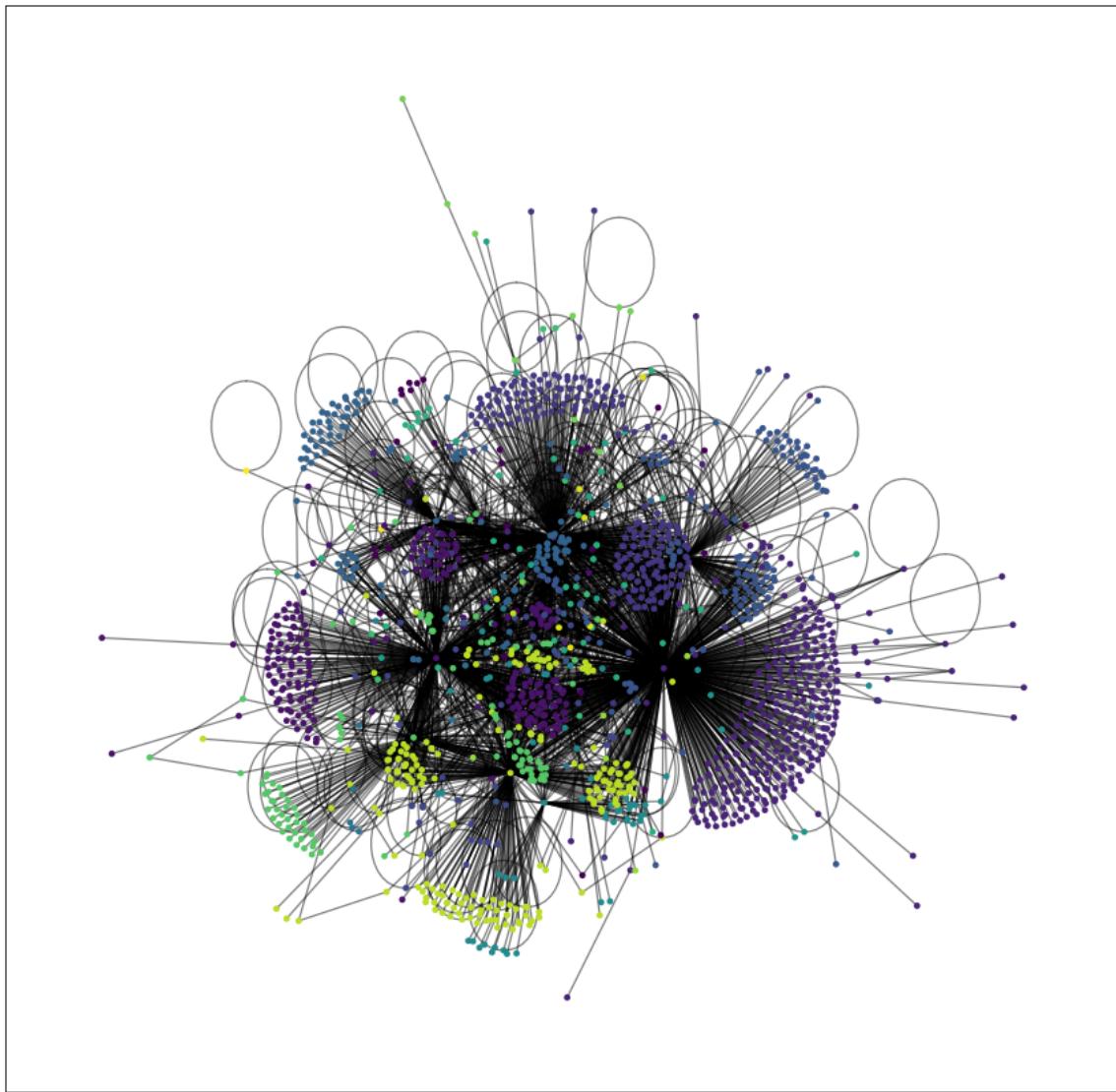
Querríamos ver ahora que comunidades existen en el grafo bajo análisis. Sabemos que cuando decimos comunidad nos referimos a un conjunto de vértices con muchas aristas hacia los vértices del conjunto, y pocas hacia afuera del mismo. Para detectar las comunidades del grafo nos vamos a valer del algoritmo de Louvain, un algoritmo greedy que ejecuta en  $O( E \log V )$ . Tiene como particularidad que nos permite tener sub comunidades en la red, es decir permite una jerarquía dentro de la comunidad.

Aplicando dicho algoritmo sobre la red obtenemos los siguientes resultados, donde cada nodo esta pintado del color correspondiente a su comunidad:



El algoritmo nos indica que hay 34 comunidades en el grafo, donde la comunidad mas grande tiene 1607 nodos, y la mas pequeña tan solo 5. Entre las restantes, varias tienen entre 300 y 500 miembros, entre 100 y 200 y algunas menos de 100. Podemos notar que en el dibujo del grafo separado en comunidades no se logra apreciar una clara separación geométrica de comunidades.

Veamos que sucede si tomamos una de las comunidades del grafo y volvemos a aplicar el algoritmo de Louvain para dividir en subcomunidades. Tomaremos la comunidad 1, la que tenia mayor cantidad de miembros, 1607:



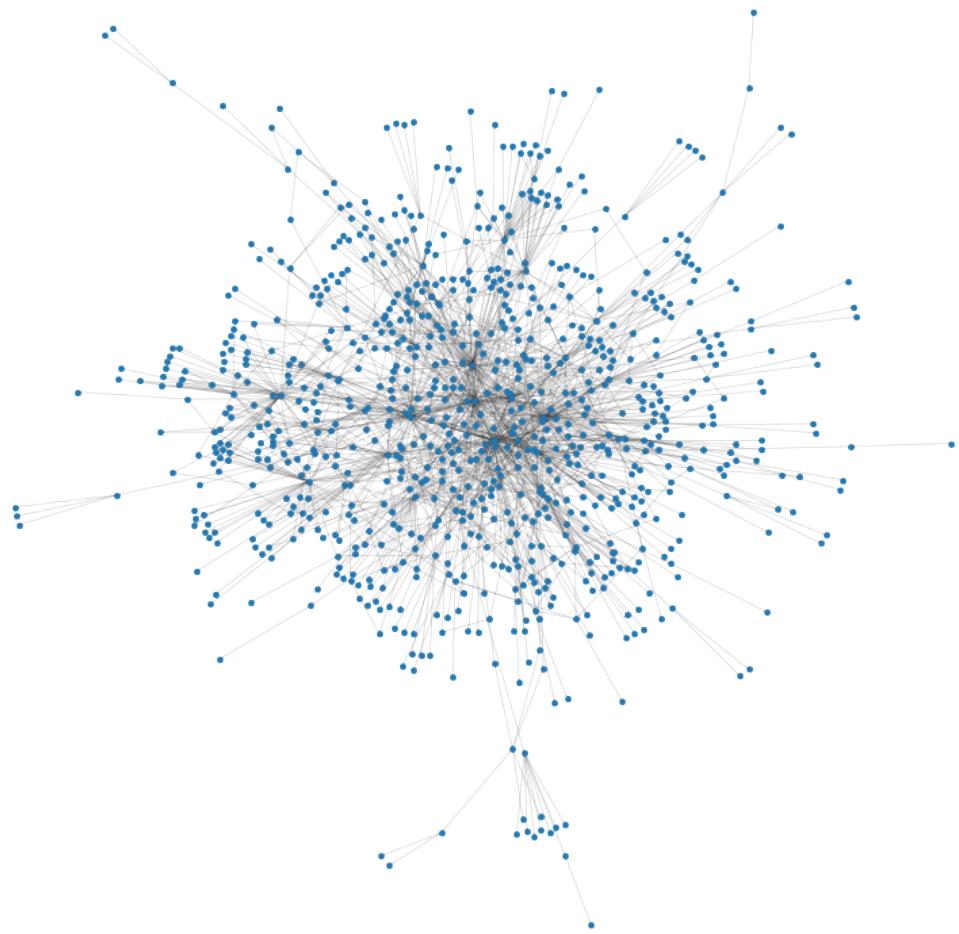
En este caso la comunidad se divide en 20 nuevas subcomunidades, donde la mas grande tiene 322 miembros y la mas pequeña tiene solo 2. En este dibujo si podemos apreciar una separación mas clara de las subcomunidades. Aun se ven algo mezcladas, pero claro, son unos cuantos nodos los que vemos en el dibujo.

## 7. Motifs

Calcularemos los motifs de hasta 4 nodos de una de las comunidades encontradas en la sección anterior. Esta comunidad es la numero 4 y posee 864 nodos. No elegimos una subred mayor debido al alto tiempo que tomaba ejecutar el análisis con redes mayores.

¿Que es un motif? Es un patrón recurrente y significante de interconexiones de nuestro grafo y su importancia radica en que nos permite entender como funciona la red y predecir comportamiento en determinadas situaciones. Sabemos que los motifs que aparecen más veces en una red real que en una aleatoria tienen una funcionalidad significativa dentro de la red.

Cabe destacar que buscamos motifs de hasta 4 nodos, y no mas, debido a que la cantidad de nodos y aristas de la subred genera que calcular un mayor valor sea muy costoso en termino de tiempo. Por otro lado, para calcular los motifs del grafo debemos eliminar los self loops, es decir eliminamos aquellas aristas que tengan un mismo nodo como origen y destino.



Buscamos entonces los motifs de hasta 4 nodos para la subred, aplicando las funciones pertinentes obtenemos lo siguiente:

[52451, 493, 495023, 1527506, 7772, 82167, 4965, 100]

Para 4 nodos hay 8 tipos de patrones o motifs diferentes. Estos valores indican cuantas veces apareció cada uno en la subred.

Lo siguiente que queríamos calcular es el promedio y desvío estándar de los motifs de una red de baseline. Para esto calculamos motifs en varias redes aleatorias utilizando configuration model y luego para cada motif obtenemos el promedio y desvío estándar. Se realizaron 10 iteraciones diferentes para calcular estos patrones (el alto tiempo que lleva cada iteración imposibilitó hacer más de ellas).

Vemos primero el promedio, es decir el promedio de veces que apareció cada patrón en la red:

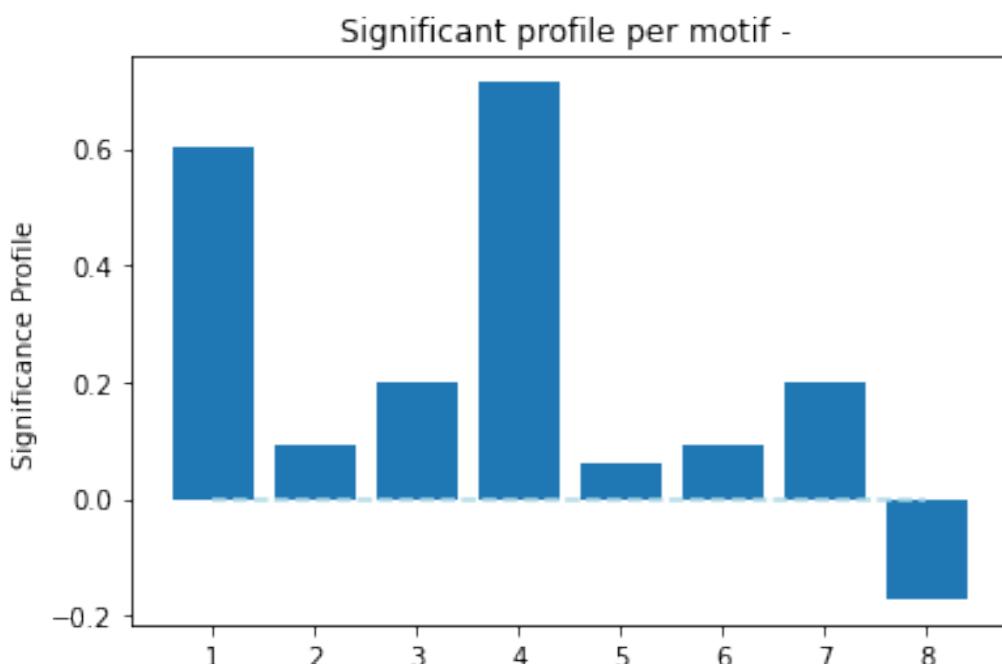
[41693, 460, 448546, 932277, 7058, 75840, 4047, 170]

Cada uno de estos valores corresponde a la misma posición de cada motif de la lista presentada mas arriba. Al ser redes aleatorias, vemos que en algún caso los valores difieren de los obtenidos con nuestra red, pero en varios casos se asemejan bastante

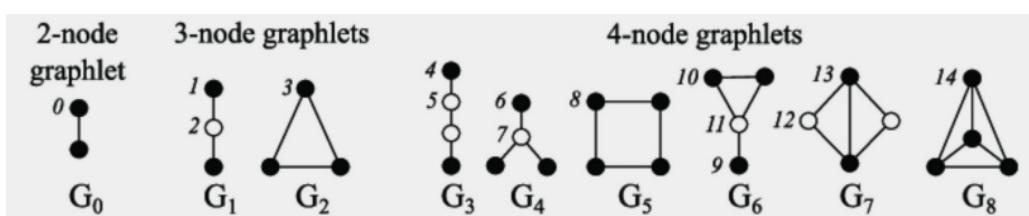
Vemos ahora los desvíos estándar para cada motif, por supuesto son 8 valores, uno para cada motif:

[1514, 30, 19514, 70565, 974, 5913, 392, 34]

Finalmente utilizando los motifs obtenidos, los promedios y desvíos estándar podemos calcular el SP, es decir el significance profile, el cual visualizamos en el siguiente gráfico:



Para tratar de entender estos resultados, podemos observar el siguiente dibujo:



Para tratar de entender estos resultados, podemos observar el siguiente dibujo: En esta imagen se ven representados los tipos de motifs que incluyen hasta 4 nodos, vemos que hay 8 tipos, misma cantidad que hablamos encontrado en nuestro grafo. Cada motif encontrado en la primera parte del ejercicio corresponde su forma a la imagen recién presentada.

Miramos ahora el grafico de SP y nos preguntamos que significa. EL SP es un vector de perfil de importancia, y se obtiene normalizando la importancia estadística de cada motif. Sabemos que los motifs que aparecen mas veces en una red real que en una aleatoria tienen una funcionalidad significativa dentro de la red. Aquellos motifs que en nuestro grafico de SP tienen un valor positivo alto aparecen en nuestra red mas veces que en nuestras redes aleatorias, tienen una significativa importancia. En cambio, los que tienen un SP negativo aparecen menos veces en la red original que en las aleatorias. Podemos contrastar entonces los valores obtenidos en el grafico de SP con el dibujo de cada motif.

Lo primero que notamos es que hay un único motif que aparece menos veces en la red original que en las aleatorias, se trata del motif que corresponde a 4 nodos todos conectados entre ellos. Si miráramos cuidadosamente nuestra red veremos que esto es real, no podemos encontrar fácilmente 4 nodos conectados entre ellos. Esto tiene sentido si recordamos la sección del Análisis Inicial, cuando habíamos determinado que nuestra red es poco completa, en base a que tiene bastantes puntos de articulación y puentes globales.

Por otro lado, vemos que hay dos tipos de motifs que parecen tener mucha importancia en nuestra red: con poca sorpresa vemos que estos motifs son el 0, dos nodos conectados, y el 3, 3 nodos conectados en linea. Decimos que no nos sorprende por lo mismo que no nos sorprendió que el motif 8 fuera poco importante. Claramente se ve a simple vista en el dibujo del grafo que hay muchas apariciones de estos motifs, ademas sustenta la teoría de que el grafo esta poco completo. La lógica indicaría que un grafo con muchas apariciones de estos motifs seria fácil de desconectar, solo removiendo un nodo o arista que pertenezca a alguno de estos grupos en lugares claves de la red, como los bordes.

## 8. Roles

En esta sección queremos detectar los roles de la red usando el algoritmo RolX.

Un rol es un conjunto de nodos que están ubicados en posiciones semejantes en la red (similaridad topologica), y se diferencia de una comunidad en que no necesariamente los nodos de un mismo rol deben estar cerca. Formalmente decimos que un rol es un conjunto de nodos que tienen un vecindario/posición similar dentro de la red.

Queremos detectar los roles dentro de nuestra red usando el algoritmo mencionado anteriormente, RolX, que implica un método de aprendizaje no supervisado, lineal.

Vamos a seguir los siguientes pasos, primero tomamos nuestro grafo original y creamos en base a eso un extractor de features recursivo, y con esto logramos justamente extraer los features. Luego creamos un extractor de roles y le pedimos que extraiga los roles en base a los features que habíamos creado antes, es decir usamos los features para detectar los roles de la red. En nuestro caso el numero de roles es determinado automáticamente mediante un procedimiento de selección de modelos.

De esta manera ya tenemos los nodos del grafo separados por rol, pero ¿como se detecta si un AS pertenece a un rol o a otro?. El funcionamiento interno del extractor de roles arma una tabla de probabilidades donde por cada rol que tiene el grafo, figura la probabilidad de cada sistema de pertenecer a cada rol, y aquel rol que tenga probabilidad mayor a los otros roles para un dado nodo sera el rol que se le asigne. Vemos algunos ejemplos:

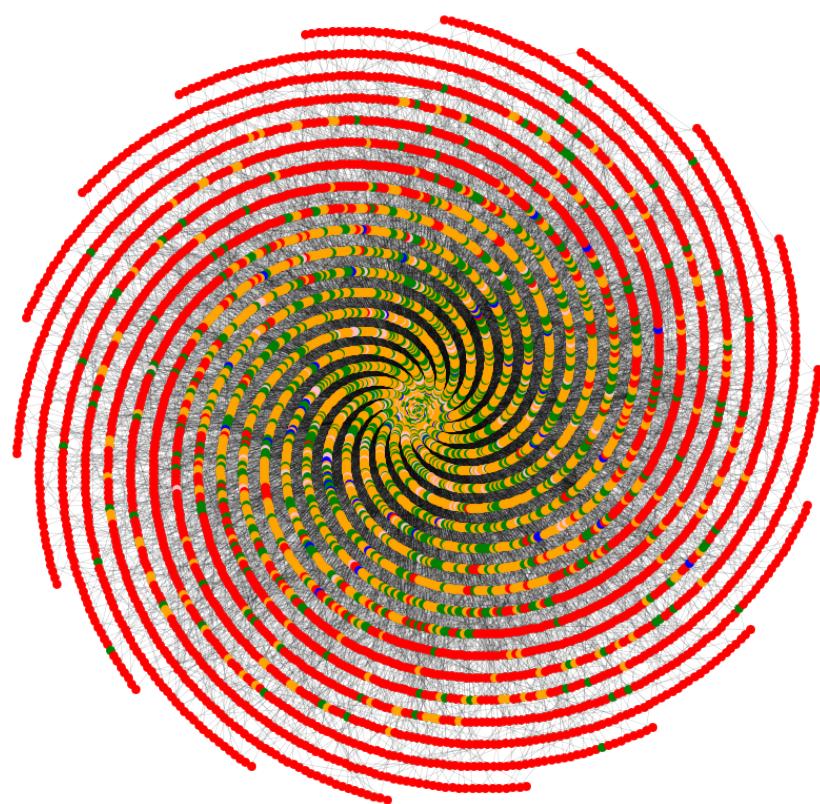
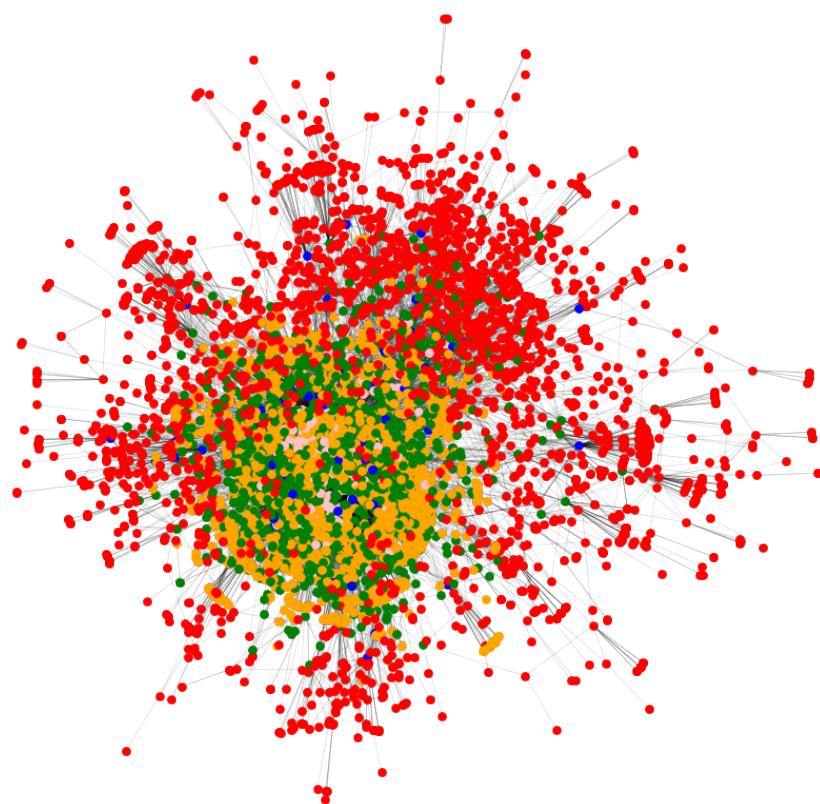
PROBABILITYS FOR EACH NODE						
	role_0	role_1	role_2	role_3	role_4	role_5
0	0.191853	0.426794	0.079853	0.079853	0.029794	0.191853
1	0.889662	0.005515	0.079009	0.014783	0.005515	0.005515
10	0.060023	0.252280	0.003846	0.676160	0.003846	0.003846
100	0.136403	0.136403	0.008739	0.573313	0.136403	0.008739
1000	0.048525	0.048525	0.757376	0.048525	0.048525	0.048525
...	...	...	...	...	...	...
995	0.028396	0.443208	0.443208	0.028396	0.028396	0.028396
996	0.048525	0.048525	0.048525	0.048525	0.757376	0.048525
997	0.048525	0.048525	0.048525	0.048525	0.757376	0.048525
998	0.028396	0.028396	0.443208	0.028396	0.443208	0.028396
999	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667

[6474 rows x 6 columns]

Vemos que en este caso habrán 6 roles diferentes, y por ejemplo el AS 0 pertenecerá al rol 1 ya que es donde tiene mayor probabilidad. A continuación se listan la cantidad de nodos por rol y que color tendrán asignado cuando grafiquemos la red:

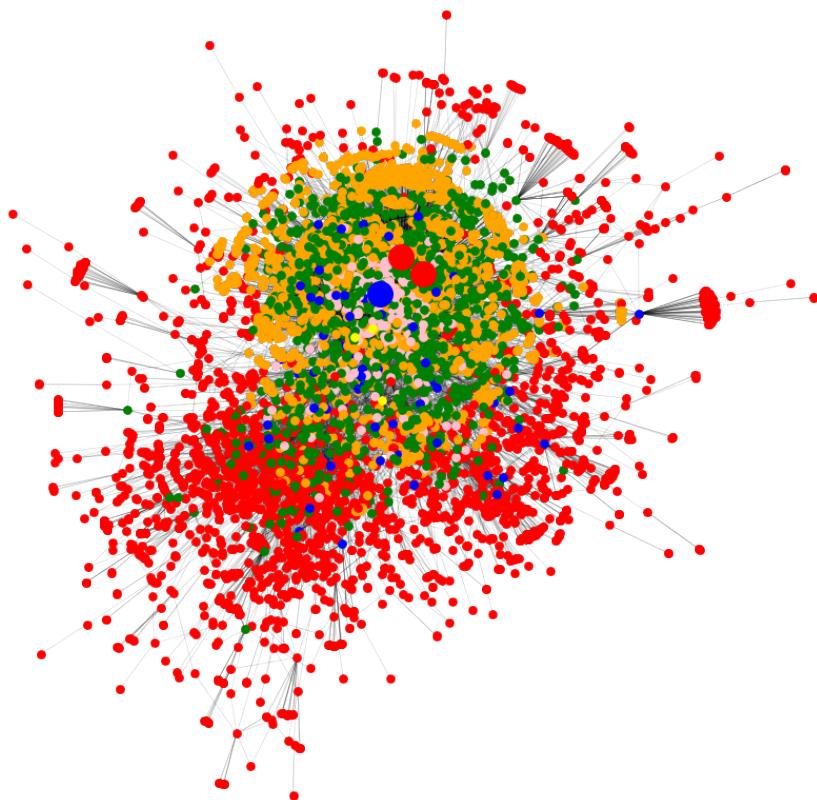
- Rol 0: 2693 nodos y color rojo.
- Rol 1: 75 nodos y color azul.
- Rol 2: 1216 nodos y color verde.
- Rol 3: 240 nodos y color rosa.
- Rol 4: 2247 nodos y color naranja.
- Rol 5: 3 nodos y color amarillo.

Presentamos ahora dos visualizaciones del grafo dividido por roles y luego intentaremos explicar que rol cumple cada nodo:



Mirando estos dibujos lo primero que podemos decir es que los nodos de color rojo parecen cumplir un rol más bien periférico en el grafo, es decir nodos topológicamente lejanos al centro y con pocas conexiones con otros nodos. Por otro lado, los nodos que parecen cumplir un rol tipo centro de estrella, es decir más tirando a centrales son los roles de color naranja y verde. Estos 3 roles mencionados son la 0, la 2 y la 4, que son las que más nodos tienen.

En la sección de centralidad habíamos comentado que los nodos más centrales del grafo son los nodos 1, 9 y 6. Querríamos saber a qué rol pertenecen dichos nodos, graficamos la red con estos tres nodos más grandes que el resto y podemos verlos aproximadamente al centro del dibujo:



Con bastante sorpresa podemos notar que el algoritmo de RolX coloca a los nodos 1 y 9, los más centrales, de color rojo! Es decir, entre los nodos que tendían a ser periféricos. Notamos igualmente que no TODOS los nodos rojos están alejados en la geometría del grafo, pero esto nos llama la atención. Por otro lado, el nodo 6, otro de los más centrales, tiene el rol de color azul, que parece un rol más tirando a central pero que se lo puede ver algo mezclado con el rojo, aunque sin ser tan periférico.

Si miramos algunos de los nodos más desconectados del grafo, como los nodos 1001, 1002 y 1003, con solo dos aristas cada uno, vemos que en general pertenecen rol 0, el más periférico del grafo.

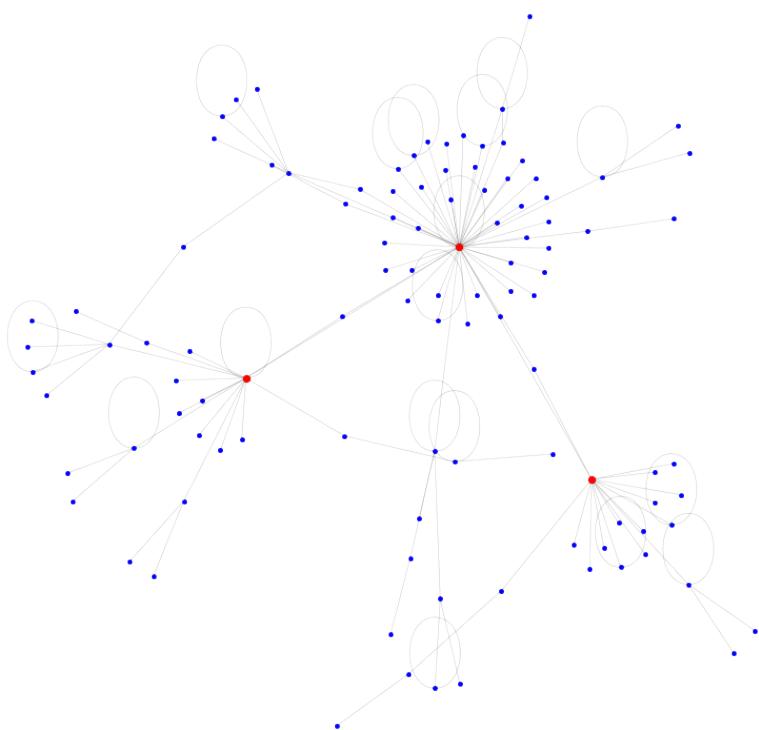
Un nodo que pertenece al rol 2, de los más centrales del grafo, es el sistema autónomo 5414, que tiene 10 aristas. Según HackerTarget, el AS 5414 pertenece al Banco de Albania. A este rol también pertenece el AS389 que corresponde al sistema autónomo llamado AFCONC-BLOCK1-AS, que son los sistemas de la fuerza aérea de los Estados Unidos.

## 9. Outbreaks

Intentaremos hacer ahora un análisis de Detección de Brotes, o Outbreaks. Lo que buscamos es que dado un proceso dinámico que se propaga a través de la red, elijamos un conjunto de nodos que detecte los procesos de forma eficiente. Esto es particularmente importante para seguridad de redes, para evitar ataques informáticos. Sería bueno que en una red de sistemas autónomos haya algunos nodos capaces de detectar un brote para poder informarlo antes que se propague por el resto de los sistemas.

Para lograr este cometido nos valdremos del algoritmo CELF, aplicando una implementación lazy para tratar de optimizar el tiempo de búsqueda. Lo que haremos será aplicar CELF para varias comunidades del grafo, variando la cantidad de iteraciones y la cantidad de nodos que queremos usar como detectores. No utilizaremos la comunidad más grande ni el grafo completo ya que a pesar de la optimización lazy, el algoritmo toma mucho tiempo en ejecutarse en redes grandes.

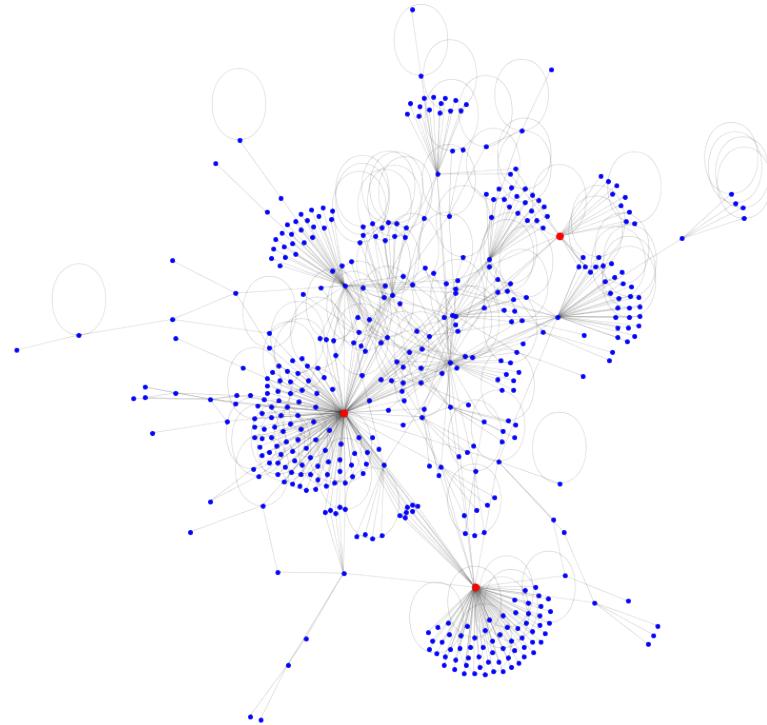
Primero iniciaremos el análisis con una comunidad relativamente pequeña, de tan solo 106 nodos, y al aplicar 1000 iteraciones con la intención de obtener 3 nodos detectores tenemos el siguiente resultado, donde los nodos rojos serán aquellos donde podemos poner un detector:



Vemos que los nodos donde poner un detector son nodos con muchas aristas (relativamente), lo cual no nos sorprende, ya que seguramente aparecerán en muchos caminos de un nodo a otro. También tienen aristas que van a nodos que tienen una única arista, o dos contando un self loop, entonces si comenzara un ataque en uno de esos nodos se detectaría rápidamente sin afectar a ningún otro AS.

Los nodos donde pondríamos detectores en esta comunidad son los sistemas automáticos 47 (University of Southern California), 49 (US National Institute of Standards & Technology) y 3893 (Headquarters, USAISC).

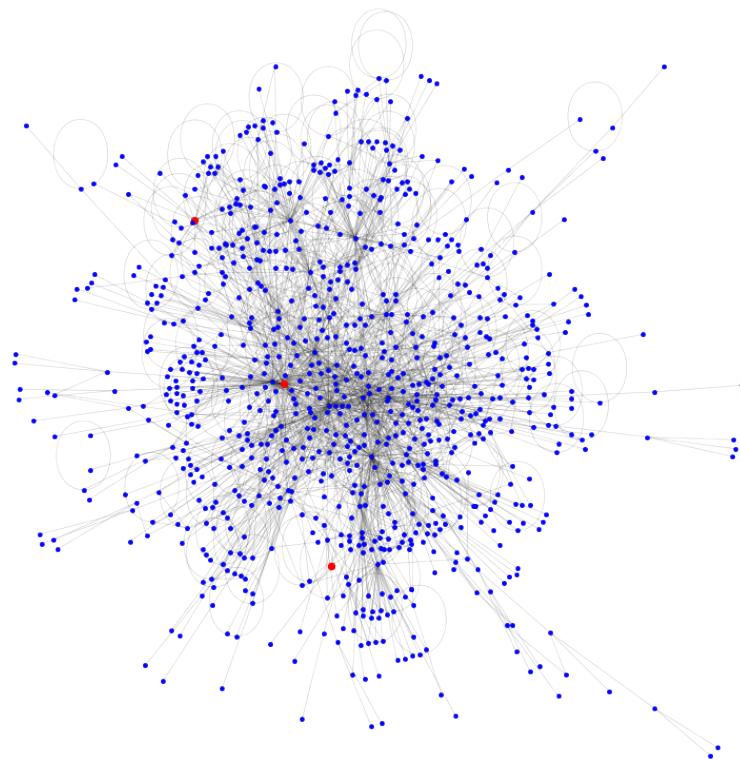
A continuación realizaremos 1000 iteraciones en el algoritmo CELF para obtener los nodos donde poner 3 detectores en una comunidad con 454 nodos.



El resultado obtenido parece ser similar a la anterior comunidad: los nodos donde poner detectores son nodos muy conectados donde muchas de sus conexiones son nodos con una única arista. Lógicamente en este caso al tener mas nodos y aristas la comunidad, pero la misma cantidad de detectores que antes, es mas peligroso que en el caso anterior que se inicie un ataque si no lo hace en uno de los nodos directamente conectados a un detector ya que puede tardar mas en llegar y en el camino potencialmente infectar varios AS.

Los AS donde poner detectores son 2 (University of Delaware), 26 (Cornell University) y 34 (University of Delaware). En este caso vemos que los AS 2 y 34 pertenecen a la misma organización pero cabe destacar que NO son el mismo AS, son efectivamente dos distintos, con el mismo dueño.

Por ultimo analizaremos una tercera comunidad con 892 nodos, aplicando 1000 iteraciones en el algoritmo CELF para buscar 3 detectores:



El resultado obtenido indica que debemos poner detectores en los sistemas autónomos 551 (Parsons Corporation), 3350 (RIPE-NCC-HM-MNT) y 5873 (DoD Network Information Center).

En este caso, si bien el sistema autónomo 551 tiene una buena cantidad de aristas, notamos que los sistemas 3350 y 5873 tienen bastante pocas. Estos datos deberían preocuparnos porque, a pesar de que los detectores están bien distribuidos geográficamente, al tener pocas aristas dos de ellos, y encima ser solo 3 detectores, en este caso la red parece estar bastante desprotegida ante ataques.

Pero, ¿realmente esta tan desprotegida la red? Si estos AS, a pesar de tener pocas aristas, aparecieran en muchos caminos mínimos... la red no sería tan vulnerable a ataques. Veamos esto valiéndonos de betweenness centrality, calculada en la sección de centralidad.

Viendo esta métrica seleccionada podemos observar que, recordando que tenemos un grafo de más de 6000 nodos, la métrica de centralidad betweenness de los AS 3350 y 5873 es bastante buena! Es decir, aparecen en más caminos mínimos que muchos otros sistemas en la red. Apenas 264 nodos en el grafo (no en la comunidad) aparecen en más caminos mínimos que el AS3350 y 625 aparecen en más caminos mínimos que el AS5873.

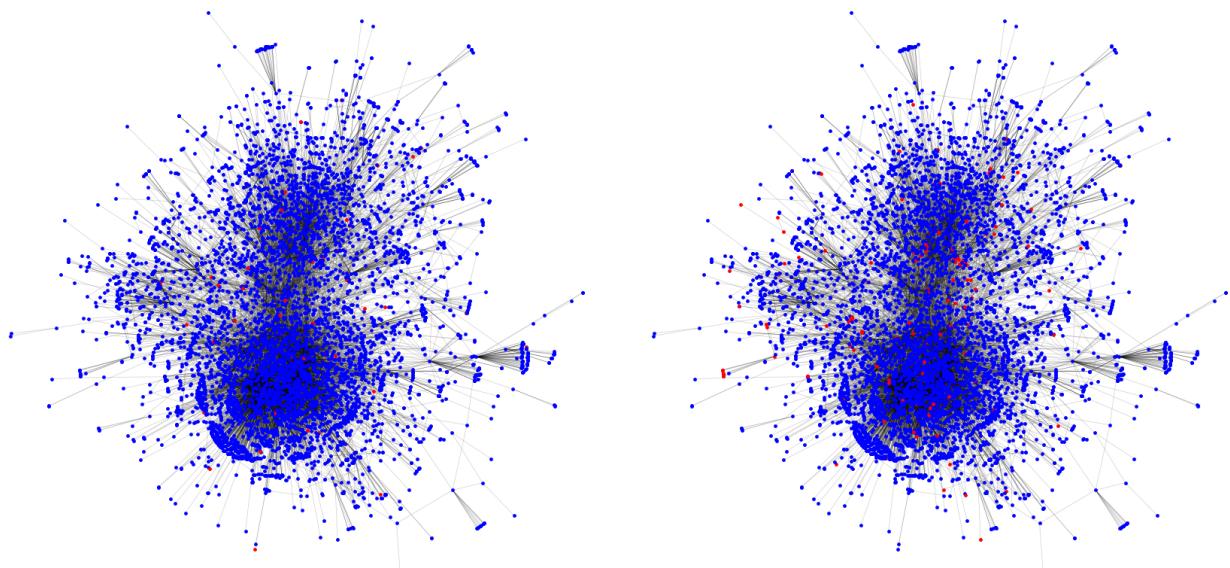
Con esto podemos decir que, a pesar de tener pocas aristas, estos sistemas autónomos serían buenos detectores para un ataque informático sobre la red.

## 10. Propagación en una red: Aplicación de Cascadas

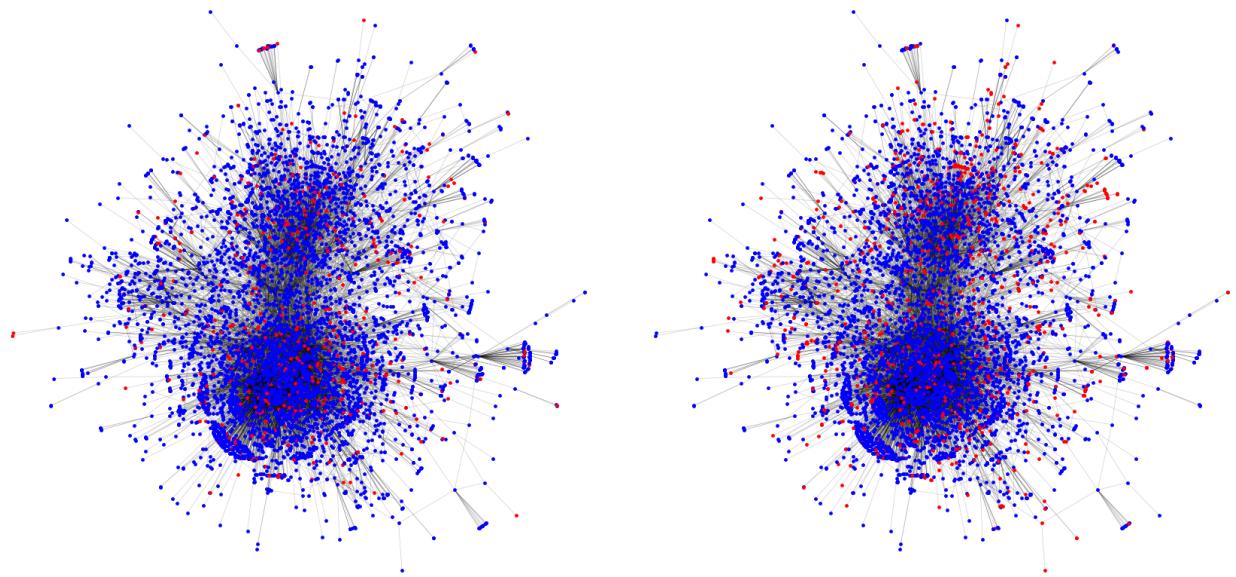
En esta sección queremos ver como puede propagarse un comportamiento a lo largo de la red. Es decir, supondremos que inicialmente los nodos tienen un determinado comportamiento B y que algunos nodos, llamados early adopters, adoptaran repentinamente el comportamiento A. Los nodos cambiaran de comportamiento siguiendo la regla del umbral, es decir si al menos la mitad de sus vecinos cambiaron de comportamiento.

Queremos saber cuantos early adopters necesitaríamos para que toda la red cambie de comportamiento, y quienes deberían ser estos early adopters. Para ello utilizaremos el grafo con todos sus nodos y aristas, y en las visualizaciones podrá verse de color azul los nodos con el comportamiento original, el B, y en rojo los nodos con el comportamiento nuevo, el A.

Inicialmente haremos que el 1 por ciento de los nodos del grafo, 66 de ellos, sean los early adopters del nuevo comportamiento, y elegiremos de forma aleatoria estos nodos. Al aplicar el algoritmo para generar el cambio de comportamiento observamos que apenas 87 los nuevos nodos que cambian de comportamiento, es decir de mas de 6000 nodos en la red solo habrán 153 con el nuevo comportamiento. En la siguiente imagen vemos en el lado izquierdo el grafo solo con los early adopters, y en el lado derecho el grafo con los nodos que cambiaron de comportamiento:

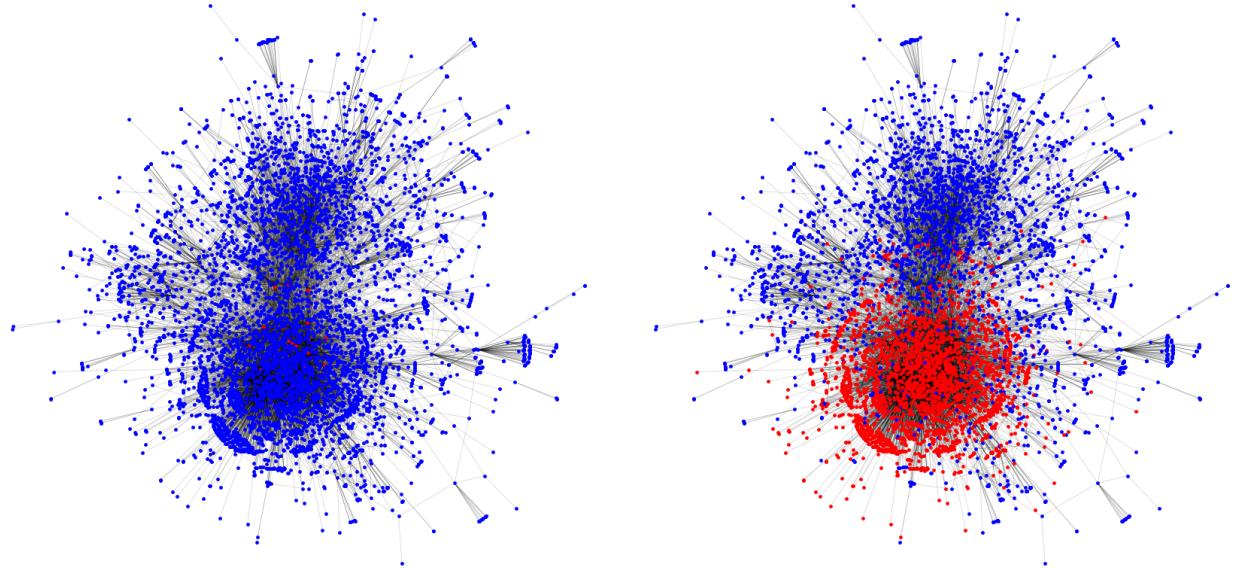


Podríamos pensar que para aumentar la propagación del nuevo comportamiento deberíamos elegir mas early adopters, así que tomamos el 10 por ciento de los nodos de la red como early adopters: 631 nodos. Observamos que al aplicar el algoritmo... tan solo 360 nodos nuevos cambiaron su comportamiento, es decir ahora hay 991 con el comportamiento A:



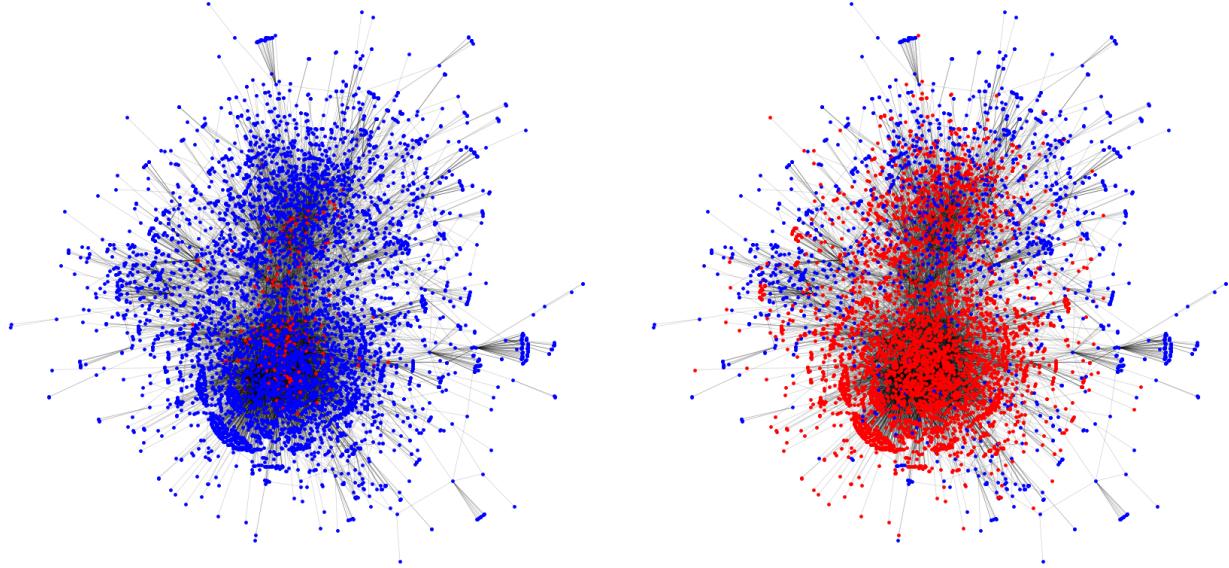
Claramente elegir nodos random para generar una cascada que cambie el comportamiento de la red no esta dando resultado, y poner aun mas early adopters no seria muy realista. Nos preguntamos entonces quienes son los nodos que comienzan las cascadas exitosas en general: la respuesta es que son los vértices mas centrales! Probaremos entonces eligiendo como early adopters aquellos mas centrales según la métrica del K-core-score.

El nodo con mayor k-core-score de este grafo tiene un k-core-score de 12, y como no es el único, elegiremos ahora como early adopters a todos los nodos de la red que tengan k-core-score = 12. Son 21 nodos que tomamos inicialmente con el nuevo comportamiento, y al aplicar el algoritmo... son mas de 3000 nodos los que cambian de comportamiento! No es suficiente para convertir a toda la red, pero parece un gran avance ya que eligiendo aproximadamente el 0.3 por ciento de la red como early adopters conseguimos convertir a la mitad de la misma.



Lo que haremos ahora sera reducir drásticamente el k-core-score de los nodos que tomamos como early adopter, con la intención de poder finalmente convertir a toda la red: aquellos con k-core-score

mayor o igual a 5 serán los early adopters, son 240 nodos de la red, alrededor del 3 por ciento. Si con diez veces menos nodos logramos convertir a media red, podríamos esperar que con el 3 por ciento se convierta la totalidad de la misma... pero esto no es lo que ocurre al correr el algoritmo: tan solo 4724 nodos serán los que ahora tengan el nuevo comportamiento, algo así como el 75 por ciento de la red:

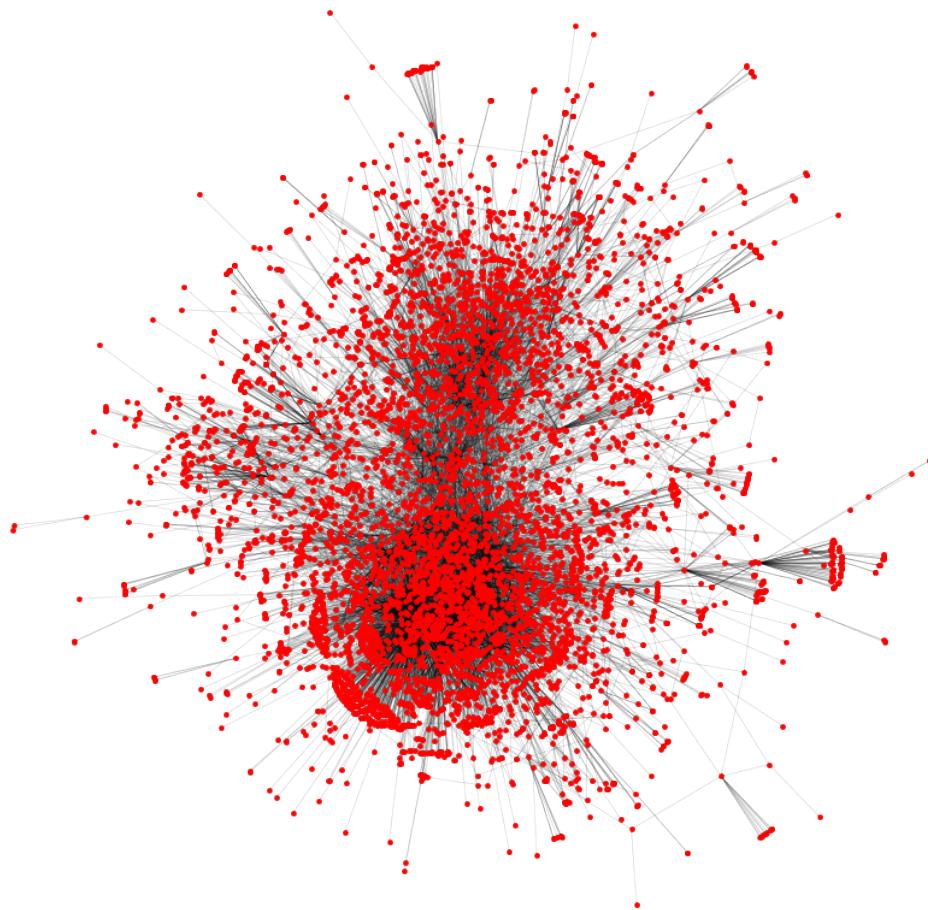


La forma de explicar esto podría ser decir que, si bien aumentamos el numero de nodos como early adopter, y encima estos son mas centrales que los que no son early adopter, al ser los nuevos nodos con el nuevo comportamiento menos centrales que los que tenían el k-core-score de 12, no alcanza para convertir a todos al nuevo comportamiento.

¿Que sigue ahora? ¿Reducir aun mas el k core score? La respuesta es negativa, en este caso, para este grafo en particular: NO hay forma de convertir a toda la red al nuevo comportamiento eligiendo como early adopter a los nodos mas centrales, literalmente hay nodos muy poco centrales que, por la geometría de la red, nunca cambiaran su comportamiento a menos que sean elegidos como early adopter, y si ellos son elegidos como early adopters por su k core score, todos los nodos de la red lo serán y el análisis no tendría sentido.

Un ejemplo del que no mostraremos visualización pues ya fueron bastantes es que poniendo un k core score mayor o igual a 2 para los early adopter, inicialmente hay 4023 con el nuevo comportamiento y luego de aplicar el algoritmo hay 6418... pero quedan 56 que se niegan a cambiar. Ya poniendo k core score de 1 el 100 por ciento de los nodos serian early que es el caso del párrafo anterior.

¿Esto quiere decir que es imposible hacer que todos los nodos cambien de comportamiento? No, podemos hacerlo con el método que descartamos inicialmente: usando early adopters random, tomando una cantidad muy alta de ellos, lo cual claramente no es aplicable a la vida real, pero únicamente para no dejarnos ganar por el grafo. Si tomamos 4523 nodos random como early adopters lograremos que al aplicar el algoritmo el 100 por ciento de la red haya cambiado de comportamiento:



Finalmente logramos ver el grafo con nodos solamente rojos que era lo que tanto anhelábamos.

¿Que conclusiones podemos obtener del análisis realizado? Nos sirve para reforzar la teoría que venimos manteniendo desde las primeras secciones: el grafo está poco conectado, por eso es tan difícil generar una cascada a través del mismo, a tal punto que para cambiar el comportamiento de toda la red necesitamos un altísimo numero de early adopters.