

Tarea 3: No somos nada, hola Internet

Manuel Figueroa

201104513-0

`manuel.figueroa@alumnos.usm.cl`

Claudio Galaz

201073598-2

`claudio.galaz@alumnos.usm.cl`

20 de Junio de 2014

1. Desarrollo

1.1. Usando Open Visual Traceroute

Utilizando Open Visual Traceroute, escriba las direcciones y observe lo que sucede. Investigue y explique porque los paquetes toman las rutas que aparecen en la herramienta. ¿Cómo viajan los paquetes de un continente a otro? Liste los enlaces internacionales que tiene Chile para conectarse a Internet.

Los routers y computadores del planeta no están conectados directamente, sino que forman una gran red de enlaces (cableadas) por todo el mundo, por ello, para enviar datos a una dirección que se encuentra alojada en algún lugar lejano es necesario pasar por distintos routers. Por ejemplo, en el caso de la solicitud de páginas web, es necesario pasar por un servidor que contiene la dirección IP a la cual está asignada el dominio solicitado, y ello lo resuelven unas bases de datos llamadas DNS, quienes pueden guardar muchas direcciones IP para un mismo nombre de dominio.

Para esta experiencia se usa Open Visual Traceroute, que consiste en un programa para realizar *traceroute* con una interfaz gráfica cómoda para el usuario. El *traceroute*, mediante el protocolo ICMP (Internet Control Message Protocol), sigue la ruta de un paquete a través de los distintos saltos que hace. Los **saltos** son los routers, computadores o cualquier cosa que se encuentre entre la fuente y el destino.

Cada paquete tiene un campo llamado TTL (time to live), o tiempo de vida, que indica la cantidad de saltos que un paquete puede viajar antes de ser descartado, y el cual es seteado en la capa de aplicación.

En cada salto, el número de TTL va disminuyendo en 1, y si el router de destino no es encontrado en ninguno de los saltos, el TTL se convierte en cero y el router bota el paquete, notificando al que envió originalmente. Típicamente el valor de TTL es 30.

Algunas consideraciones sobre el *traceroute*:

- Solo se ve la dirección de los paquetes que están siendo enviados. Esto no quiere decir que los que sean enviados de vuelta usen la misma ruta. El tráfico podría ser enviado por otra red sin siquiera saberlo a menos que se haga un traceroute en ese sentido.
- Lo que se ve identificado en el seguimiento es el puerto por el que los paquetes entraron al router. Nunca se es capaz de ver cuánto tráfico sale de ese router; sólo se puede ver cómo entra al router y luego cómo entra el siguiente paquete.
- Existen maneras de identificar routers con su respectiva localización. Esta información puede resultar muy importante para identificar rutas con loops y ruteos poco óptimos.
- A veces los paquetes del traceroute son bloqueados, lo que es debido a asuntos de seguridad o protección contra DDoS.

Entonces, analizando lo obtenido con el programa se puede observar que en la mayoría de los casos el paquete sale del país; sólo con <http://moodle.inf.utfsm.cl> el paquete se queda en Chile. Esto sucede porque la plataforma Moodle del departamento es frecuentemente solicitada dentro de Chile, entonces no hay necesidad de ir a buscar la dirección a un servidor DNS fuera del país porque ya está guardada en el caché del servidor nacional. Al final llega a Valparaíso puesto que el Moodle del departamento de informática se encuentra alojado en Casa Central.

Para `http://cime.cl` es necesario ir a buscar la DNS a otro lugar fuera de Chile, y luego va a Estados Unidos, donde el mayor tiempo de DNS Lookup se ocupa en New York y además es donde probablemente esté alojado el sitio. Hay que destacar el hecho que de los cinco sitios analizados es el que más tiempo ocupa en el DNS lookup.

Con Google ocurre algo totalmente distinto. Al ser el sitio más solicitado en Chile¹, claramente el DNS lookup ocurre dentro de Chile, por lo que el paquete es enviado inmediatamente a los Estados Unidos que es donde se encuentra alojado.

Con Wikipedia el paquete viaja hasta España, y luego es enviado a Francia dos veces, donde pasa una gran cantidad de tiempo en el DNS lookup, buscando por la dirección a la cual corresponde el nombre. Luego de eso pasa por un router en New York antes de llegar a San Francisco.

Por último con la embajada de Australia, luego de viajar varios kilómetros hacia España y además hacer saltos por Europa, el paquete llega a Japón, donde luego de 10 segundos (timeout seteado por defecto en la aplicación) no muestra ningún otro salto. Aunque con un poco de demora puesto que debe dar la vuelta completa al planeta para llegar a Australia (donde debería estar alojado el sitio), el sitio carga perfectamente, lo que se debe a que el router en Japón permite el tráfico HTTP pero no los paquetes del protocolo ICMP, a quienes los hace esperar indefinidamente o simplemente los desecha, causando timeout.

La evidencia con los screen shots se encuentra en la sección de Anexos.

Con respecto a cómo viajan los paquetes de un continente a otro, lo hacen mediante cables submarinos y en la actualidad existen tres enlaces internacionales de fibra óptica en Chile, los que sirven para comunicar a nuestro país con el resto del mundo. Aquellos cables son:

- Pan American (PanAm), en Arica, conectado con Perú, Ecuador, Panamá, Colombia, Venezuela y las Islas Virginius.
- South America-1 (SAm-1), encontrado en Arica y Valparaíso, conectando la mayoría de los países de Sudamérica con América Central. Observar la figura en los anexos.
- South American Crossing (SAC)/Latin American Nautilus (LAN), que se encuentra en Valparaíso y conecta a Perú, Colombia, Panamá, Venezuela, las Islas Virginius, Brasil y Argentina.

Hay que destacar que para Julio de 2016 estará listo el South America Pacific Link (SAPL), que se encontrará en Arica y Valparaíso, teniendo conexión con Oahu en Hawaii. Esto permitirá una transmisión de paquetes más ágil con el resto del mundo.

Imágenes de los enlaces anteriores pueden ser consultados en la sección Anexos.

En la actualidad el continente antártico es el único que no tiene conexión por cable con el resto del mundo, realizando todo tipo de comunicaciones vía satélite. Esto es debido al tipo de cable que es necesario desarrollar para soportar las temperaturas de Antártica. En la actualidad el tipo de cable submarino usado para conectar internet entre continentes se compone de (enlistados desde la capa más exterior y hacia el interior):

- Polietileno.
- BoPET.

¹ “Revelan mapa con los sitios más visitados por país: Google es el favorito en Chile.” <http://goo.gl/vXLb9V>

- Alambres de acero trenzado.
- Barrera de aluminio resistente al agua.
- Policarbonato.
- Tubo de cobre o aluminio.
- Vaselina.
- Fibras ópticas.

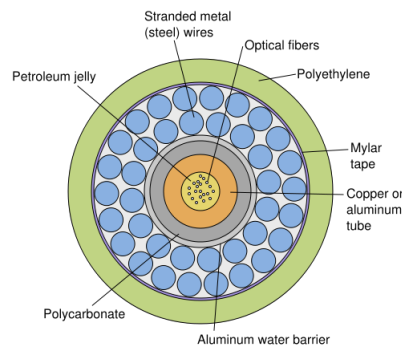
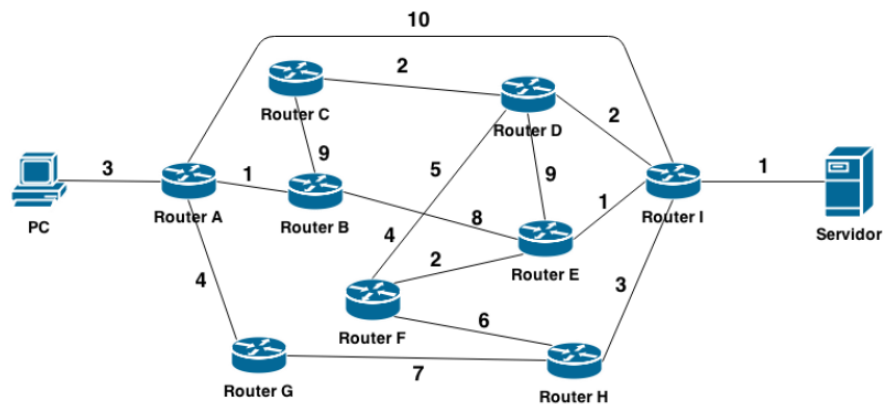


Figura 1: El cable submarino de comunicaciones y una mirada a través de un corte transversal

1.2. Cálculo de rutas: Algoritmo vector-distancia

Se dispone de la siguiente red a menor escala compuesta por 9 routers. El PC desea acceder a los archivos que se encuentran en el servidor.



Utilizando el Algoritmo Vector-Distancia, encuentre las tablas de costos de todos los routers.

En Python se realizó un script que aplica el algoritmo vector-distancia para encontrar las distancias mínimas entre nodos. El script imprime en consola las matrices que muestran la distancia entre los distintos con cada iteración.

```
Matriz original:

      A    B    C    D    E    F    G    H    I
A [  0.   1.  inf  inf  inf  inf   4.  inf 10.]
B [  1.   0.   9.  inf   8.  inf  inf  inf  inf]
C [ inf   9.   0.   2.  inf  inf  inf  inf  inf]
D [ inf  inf   2.   0.   9.   4.  inf  inf   2.]
E [ inf   8.  inf   9.   0.   2.  inf  inf   1.]
F [ inf  inf  inf   4.   2.   0.  inf   6.  inf]
G [  4.  inf  inf  inf  inf  inf   0.   7.  inf]
H [ inf  inf  inf  inf  inf   6.   7.   0.   3.]
I [10.  inf  inf   2.   1.  inf  inf   3.   0.]
```

Figura 2: Matriz de distancias original.

La matriz original muestra la distancia entre cada nodo y sus adyacentes, aquellos que no son adyacentes tienen un valor infinito. Es una forma simple de mostrar la matriz que tiene cada nodo, donde sólo poseen la información de las distancias entre ellos y sus adyacentes, es decir: su vector fila con información y el resto en infinito.

Teniendo lo anterior en cuenta, para la primera iteración se asume que los nodos hicieron un broadcast de sus vectores distancia y todos los nodos poseen la matriz original. En base a esto se realiza el cálculo de las distancias mínimas de la siguiente forma:

$$D_x(y) = \min\{c(x, v) + D_v(y)\} \quad (1)$$

para cada nodo $y \in N$, siendo N el conjunto de los routers.

La función de distancia asigna el valor mínimo de la suma de la distancia a cada nodo vecino con la distancia entre éste y el nodo al que se quiere llegar. Cuando se actualizan las distancias, el algoritmo revisa si alguna distancia cambió, de ser así se realiza otra iteración hasta lograr llegar a los valores mínimos posibles.

La siguiente figura muestra las iteraciones sobre la matriz inicial:

```

Iteracion # 1 :

      A      B      C      D      E      F      G      H      I
A [  0.    1.   10.   12.    9.  inf    4.   11.   10.]
B [  1.    0.    9.   11.    8.   10.    5.  inf    9.]
C [ 10.    9.    0.    2.   11.    6.  inf  inf    4.]
D [ 12.   11.    2.    0.    3.    4.  inf    5.    2.]
E [  9.    8.   11.    3.    0.    2.  inf    4.    1.]
F [ inf  10.    6.    4.    2.    0.   13.    6.    3.]
G [  4.    5.  inf  inf  inf   13.    0.    7.   10.]
H [ 11.  inf  inf    5.    4.    6.    7.    0.    3.]
I [ 10.    9.    4.    2.    1.    3.   10.    3.    0.]

Iteracion # 2 :

      A      B      C      D      E      F      G      H      I
A [  0.    1.   10.   12.    9.   11.    4.   11.   10.]
B [  1.    0.    9.   11.    8.   10.    5.   12.    9.]
C [ 10.    9.    0.    2.    5.    6.   14.    7.    4.]
D [ 12.   11.    2.    0.    3.    4.   12.    5.    2.]
E [  9.    8.    5.    3.    0.    2.   11.    4.    1.]
F [ 11.   10.    6.    4.    2.    0.   13.    6.    3.]
G [  4.    5.   14.   12.   11.   13.    0.    7.   10.]
H [ 11.   12.    7.    5.    4.    6.    7.    0.    3.]
I [ 10.    9.    4.    2.    1.    3.   10.    3.    0.]
Rutas minimizadas

```

Figura 3: Iteraciones para obtener distancias mínimas.

En tan sólo dos iteraciones se obtienen los valores mínimos de distancia entre nodos.

1.3. Emergencia en la red

Un barco transatlántico ha cortado el enlace entre los nodos H e I, ¿Cómo se comporta el algoritmo vector distancia en este caso? Recalcule solo los costos necesarios para resolver este problema.

Con una pequeña modificación al script anterior, se hace que cada vez que se utilice el enlace H-I en el cálculo de distancia entre dos nodos, se marque éste par de nodos.

```

#####UN BARCO CORTA LAS LINEAS ENRE H e I #####
Se desconocen los costos que pasaban por esa linea:

      A    B    C    D    E    F    G    H    I
A [  0.   1.  10.  12.   9.  11.   4.  11.  10.]
B [  1.   0.   9.  11.   8.  10.   5.  12.   9.]
C [ 10.   9.   0.   2.   5.   6.  14.  inf   4.]
D [ 12.  11.   2.   0.   3.   4.  inf  inf   2.]
E [  9.   8.   5.   3.   0.   2.  inf  inf   1.]
F [ 11.  10.   6.   4.   2.   0.  13.   6.   3.]
G [  4.   5.  14.  inf  inf  13.   0.   7.  inf]
H [ 11.  12.  inf  inf  inf   6.   7.   0.  inf]
I [ 10.   9.   4.   2.   1.   3.  inf  inf   0.]

Iteracion # 1 :

      A    B    C    D    E    F    G    H    I
A [  0.   1.  10.  12.   9.  11.   4.  11.  10.]
B [  1.   0.   9.  11.   8.  10.   5.  12.   9.]
C [ 10.   9.   0.   2.   5.   6.  14.  12.   4.]
D [ 12.  11.   2.   0.   3.   4.  16.  10.   2.]
E [  9.   8.   5.   3.   0.   2.  13.   8.   1.]
F [ 11.  10.   6.   4.   2.   0.  13.   6.   3.]
G [  4.   5.  14.  16.  13.  13.   0.   7.  14.]
H [ 11.  12.  12.  10.   8.   6.   7.   0.   9.]
I [ 10.   9.   4.   2.   1.   3.  14.   9.   0.]
Rutas minimizadas

```

Figura 4: Iteraciones para obtener distancias luego del accidente.

Como se observa en la matriz de la figura 4, las distancias que fueron marcadas por usar el enlace H-I fueron cambiadas por “Inf”, dado que sin el enlace disponible se desconoce la distancia mínima. Luego de esto se itera de nuevo para encontrar las distancias mínimas que se perdieron sin considerar dicho enlace.

2. Referencias

- Kurose, Jim. (2009). **Computer Networking: A Top Down Approach**, 5th Edition.
- TeleGeography. **Submarine Cable Map** <http://www.submarinecablemap.com/>

3. Anexos

3.1. ejercicio 2.py

El script genera todo lo necesario para las preguntas 2 y 3, obteniendo la distancia mínima entre los nodos y arreglando el error cuando se corta el enlace entre H e I.

```
# -*- coding: cp1252 -*-
from numpy import *

def matriz_1():
    a = zeros((9,9))
    a[0] = [0,1,inf,inf,inf,inf,4,inf,10]
    a[1] = [1,0,9,inf,8,inf,inf,inf,inf]
    a[2] = [inf,9,0,2,inf,inf,inf,inf,inf]
    a[3] = [inf,inf,2,0,9,4,inf,inf,2]
    a[4] = [inf,8,inf,9,0,2,inf,inf,1]
    a[5] = [inf,inf,inf,4,2,0,inf,6,inf]
    a[6] = [4,inf,inf,inf,inf,inf,0,7,inf]
    a[7] = [inf,inf,inf,inf,inf,6,7,0,3]
    a[8] = [10,inf,inf,2,1,inf,inf,3,0]
    return a

def imprimir(a):
    bla = ['A','B','C','D','E','F','G','H','I']
    print "  A   B   C   D   E   F   G   H   I"
    for i in range(9):
        print bla[i], a[i]

def rutas(a,marcas):
    anterior = copy(a)
    contador = 0
    while True:
        contador += 1
        for i in range(9):
            for j in range(9):
                for z in range(9):
                    temp = copy(a[i,j])
                    a[i,j] = min(a[i,j], (anterior[i,z]+anterior[z,j]))
                    if (z == 7 and (i==8 or j==8)) or (z==8 and (i==7 or j==7)) or marcas[i,z]== 1
                        or marcas[z,j] ==1:
                        if temp != a[i,j]:
                            marcas[i,j] = 1
        if array_equal(a,anterior):
            print "Rutas minimizadas"
            break
        print "\nIteracion #",contador,":\n"
        imprimir(a)
        anterior = copy(a)
```



```

    return a

def main():
    a = matriz_1()
    marcas = zeros((9,9))
    print "Matriz original: \n"
    imprimir(a)
    rutas(a,marcas)
    print"\n#####UN BARCO CORTA LAS LINEAS ENRE H e I #####"
    a[7,8] = float("inf")
    a[8,7] = float("inf")
    for m in range(9):
        for n in range(9):
            if marcas[m,n] == 1:
                a[m,n]=float("inf")
    print "Se deconocen los costos que pasaban por esa linea:\n"
    imprimir(a)
    rutas(a, marcas)

if __name__ == '__main__':
    main()

```

3.2. Screenshots de Open Visual Traceroute

Para una mayor resolución se puede consultar el directorio fuente del informe.

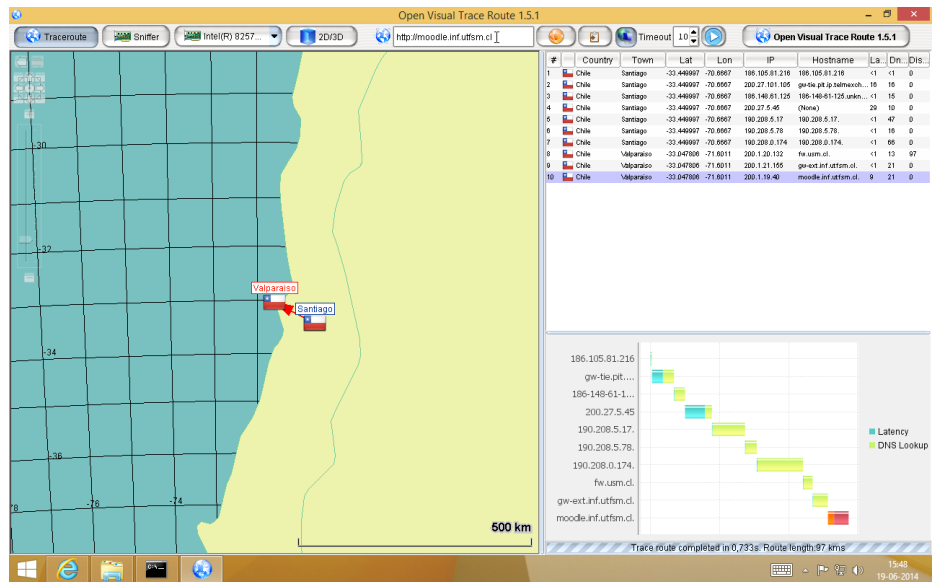
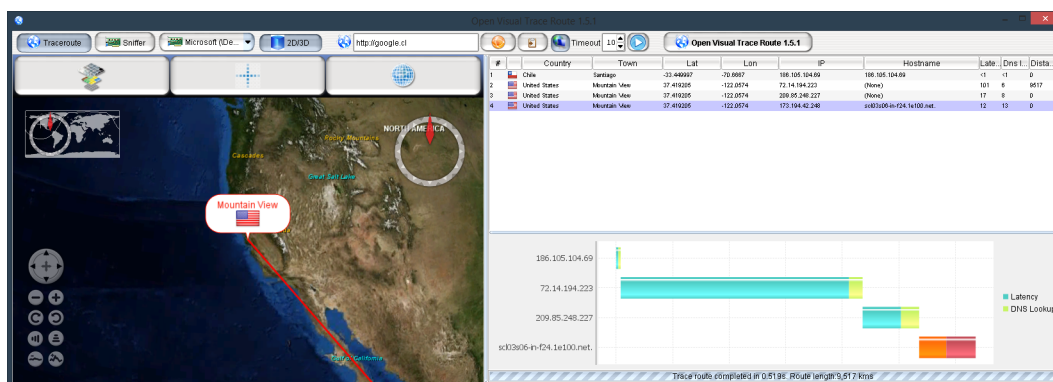
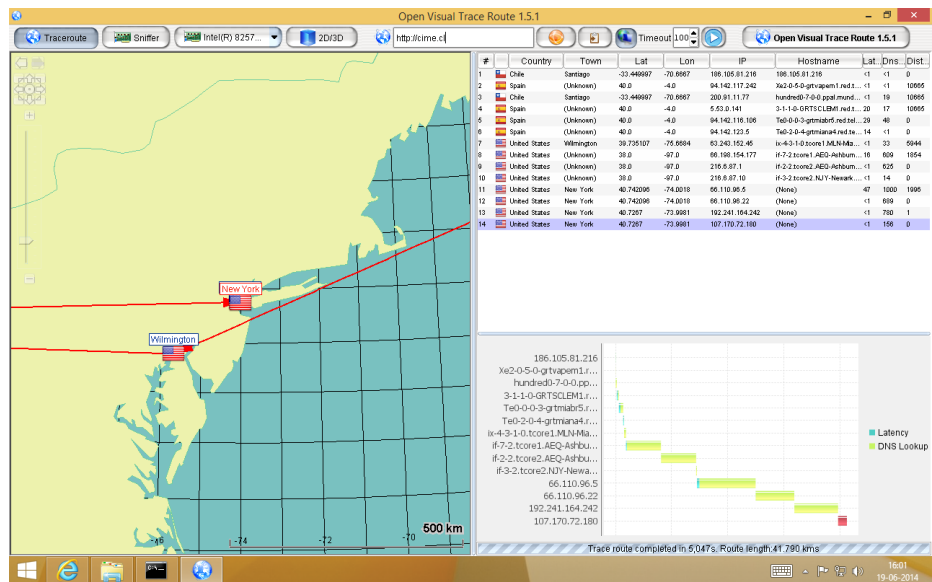


Figura 5: Traceroute para el sitio <http://moodle.inf.utfsm.cl>



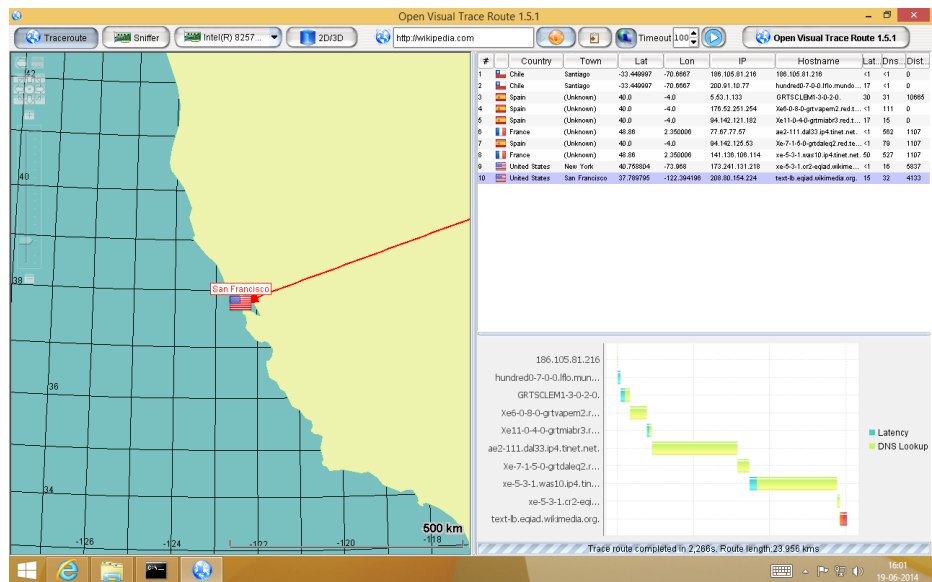


Figura 8: Traceroute para el sitio <http://wikipedia.com>

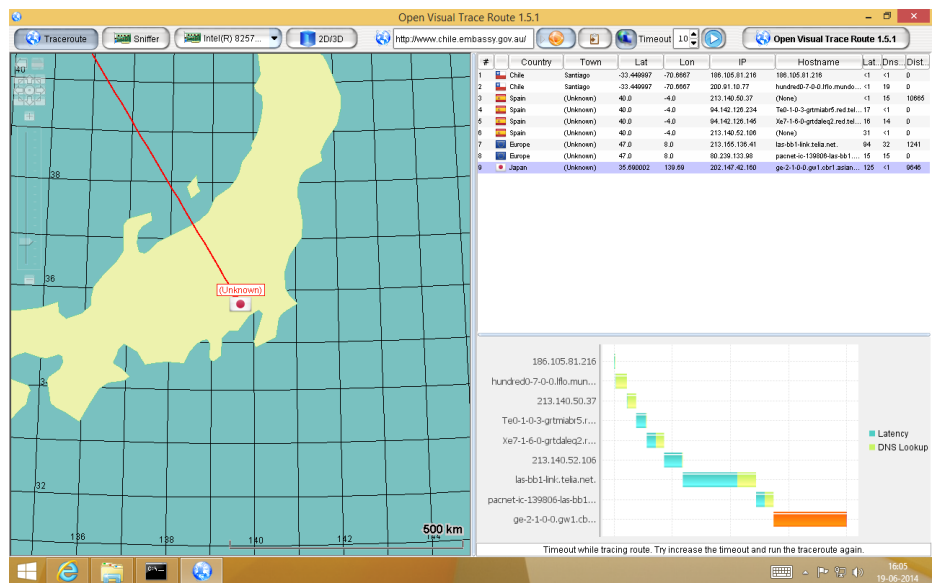


Figura 9: Traceroute para el sitio <http://www.chile.embassy.gov.au>

3.3. Cables submarinos en enlace con Chile



Figura 10: Conexiones de South America-1



Figura 11: Conexiones de Pan American



Figura 12: Conexiones de South American Crossing/Latin American Nautilus

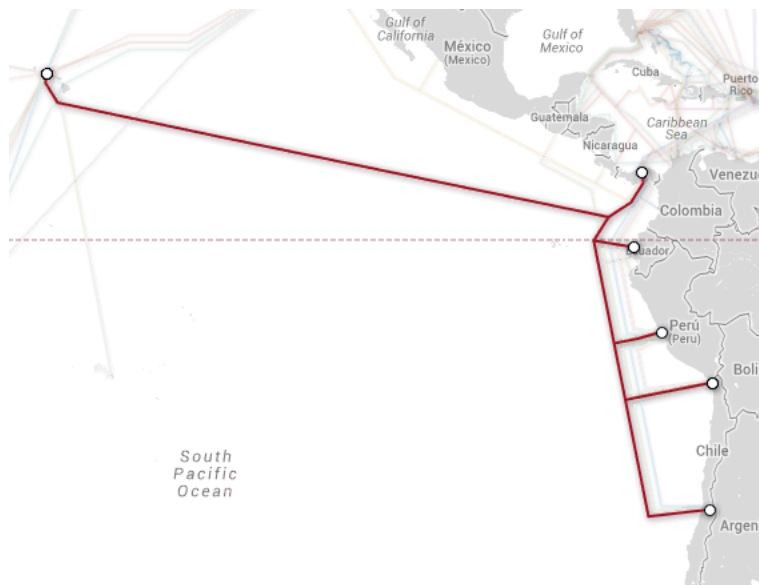


Figura 13: Conexiones de South America Pacific Link (en proceso)