

Fragmentación y Mezcla de redes ad-hoc usando el protocolo ANTop

Fragmentation and assimilation of ad-hoc networks using the ANTop protocol

Pablo Torrado

Facultad de Ingeniería, Universidad de Buenos Aires, Argentina

ptorrado@cnet.fi.uba.ar

Recibido: 06/08/18; Aceptado: 19/11/18

Abstract— The ANTop (Adjacent Network Topology) is a distributed and decentralized protocol for ad-hoc networks. It is based on a hypercube address space where the nodes names are mapped to assure their adjacency, allowing greedy routing protocols to find the optimum route. A distributed mechanism, to match real names to the network address, is implemented by distributed hashing tables. In this work we address the partition and assimilation networks problems, to complete the functionalities of ANTop, and we also provide a complete implementation for Linux systems.

Keywords: ad-hoc networks, IoT, Internet

Resumen—ANTop (Adjacent Network Topology) es un protocolo distribuido y descentralizado para redes ad-hoc. Está basado en un hipercubo para el espacio de direccionamiento, donde se asegura la adyacencia de las direcciones de los nodos, lo cual permite que protocolos de ruteo gluttones encuentren la ruta óptima. Un mecanismo distribuido, que establece una analogía entre los nombres reales y las direcciones de red, es implementado usando tablas distribuidas de hash. En este trabajo nosotros estudiamos los problemas de partición y mezcla, con el fin de completar las funcionalidades de ANTop, y proveer una implementación en sistemas Linux.

Palabras clave: redes ad-hoc, IoT, Internet.

I. INTRODUCCIÓN

En el presente trabajo se presenta nuevas funcionalidades al protocolo de ruteo ANTop (Adjacent Network Topology) para redes ad-hoc, desarrollado en los trabajos [1], [2], [6], [11], [13]. El alcance de este trabajo es responder a la aplicación del protocolo de ruteo ANTop ante las problemáticas de fragmentación y mezcla de redes, muy común en redes ad-hoc y también de sensores. A parte de ANTop, en la actualidad existen protocolos de ruteos para redes ad-hoc como B.A.T.M.A.N. (Better Approach To Mobile Ad-hoc Networking) [5], [7] y BABEL [3], [4], [8]–[10], que cuyas diferencias básicas con nuestra propuesta reside en que sus algoritmos de ruteo se basan en un conocimiento global de la estructura de la red (ejemplo cada nodo calcula el próximo salto hacia el nodo destino, para cada destino). Para ello, dichos protocolos deben de enviar paquetes a la red de tal forma de descubrir cuál es la ruta mas eficiente de un nodo hacia al otro como destino, disminuyendo el ancho de banda útil disponible en la red. En cambio, ANTop propone un direccionamiento basado en un hipercubo, definiendo así la estructura de la red sin la necesidad del envíos de paquetes de reconocimiento de red, lo que permite ahorra energía y capacidad de los canales para la transmisión de la información útil.

En este artículo se propone presentar las características principales del protocolo ANTop en primer lugar, para

dar el contexto de su funcionamiento; y luego detallar las problemáticas de fragmentación y mezcla con sus algoritmos para mitigarlos.

II. EL PROTOCOLO ANTOP

Como se menciona antes, ANTop es un protocolo diseñado para redes ad-hoc, con un esquema descentralizado, ya que todos los nodos conectados tienen igual jerarquía, pudiendo cumplir las mismas funciones, como ser la asignación de direcciones a los nuevos vecinos, resolución de direcciones a partir de un identificador universal o efectuar el ruteo de paquetes. Es por esta razón que su dirección, dependiente de su ubicación en la red, cambiará frecuentemente; y por lo tanto se vuelve imprescindible contar con un identificador universal del nodo. Este identificador, sin embargo, no será suficiente para poder rutear un paquete hacia su destino. En respuesta a éste requerimiento, surge el concepto de *Distributed Hash Tables* (DHT), es decir, Tablas de Distribuidas de Hash. Incorporando esta abstracción a la capa de red, ANTop define una técnica de ruteo indirecto, en la cual nodos llamados *Rendez Vous* son responsables de guardar la dirección de red de otros nodos, asociándola con su identificador universal. Como se dijo anteriormente, cualquier nodo que se conecte a la red ad-hoc podría actuar como *Rendez Vous*.

II-A. Ruteo Indirecto

Cada nodo cuenta con tres direcciones. La primera es la dirección universal U , la cual es conocida por cualquier otro nodo que quiera establecer una comunicación. Al ser totalmente independiente de la capa de red, puede ser cualquier tipo de identificador. La segunda dirección de un nodo, la relativa \mathcal{R} , es aquella dependiente de la ubicación en la topología de red y la que nos permitirá rutear los paquetes. Si el nodo se mueve, este identificador cambiará. Finalmente la tercer dirección, la virtual \mathcal{V} , se obtiene aplicando una función de Hash lineal a la U y será la dirección \mathcal{R} , que estará contenida dentro del espacio del nodo *Rendez Vous* T , quién está a cargo de proveer el identificador relativo asociado a esa dirección U . El esquema de ruteo indirecto funciona en la siguiente forma. En la figura 1 el nodo origen s , requiere comunicarse con el destino f , pero no sabe su dirección \mathcal{R} , tan sólo la U . Por esta razón, s contactara al nodo *Rendez Vous* T quien conoce la dirección \mathcal{R} correspondiente al nodo f (notar que este último se registró con T al conectarse a la red). A continuación T envía un mensaje a s , que contiene la dirección \mathcal{R} de f ,

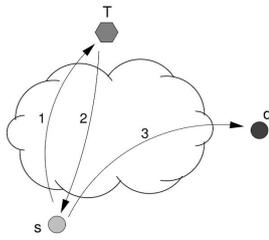


Figura 1. Modelo de ruteo indirecto.

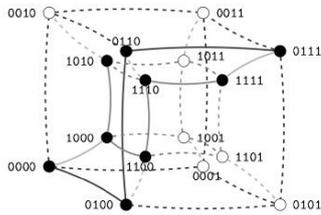


Figura 2. Direccionamiento basado en hipercubo.

y entonces de modo el nodo s puede enviar paquetes a f porque conoce su ubicación en la red.

II-B. Direccionamiento y ruteo en ANTop

El protocolo ANTop propone dos metodologías para enviar un paquete: el ruteo proactivo y el ruteo reactivo. En el primero de ellos las tablas de ruteo se construyen y se mantienen actualizadas constantemente, asegurando una ruta a cada nodo de la red, en todo momento. En el segundo esquema las rutas se generan bajo demanda y se mantienen durante un determinado lapso de tiempo. Dada una red ad-hoc, donde la posición de cada nodo puede variar con frecuencia, dependiendo de cuán rápido varían se elige el primer (poca variación) o segundo esquema (rápida variación).

Para poder realizar el ruteo, ANTop define un esquema de direccionamiento basado en hipercubos. Estos son una generalización de un cubo de tres dimensiones, a un número d . Cada nodo tiene una coordenada de valor 0 o 1 en cada dimensión, entonces el número total de nodos 2^d . Cada nodo está conectado a aquellos cuyas coordenadas varían sólo en una dimensión. En la figura 2 se muestra un hipercubo de dimensión 4. ANTop utiliza las coordenadas de cada nodo en el hipercubo para asignar a cada nodo que se conecta una dirección \mathcal{R} . Adicionalmente, el nuevo nodo podrá recibir otras direcciones, denominadas secundarias, de otros nodos ya conectados a la red, con el fin de mejorar su interconexión.

En este esquema, es fácil ver que la distancia entre dos nodos. Como vemos la distancia en el hipercubo está definida por la cantidad de dimensiones con coordenadas distintas; que se puede obtener aplicando una función XOR entre ambas direcciones. Por ejemplo la distancia entre los nodos 0100 y 0110 es 1. La distancia se puede pensar como la cantidad de nodos mínima por el que un mensaje de un nodo debe pasar para llegar a un destinatario.

Cuando un nuevo nodo está en el radio de cobertura de alguno de los ya conectados a la red, hará un pedido en modo broadcast para que le sea asignada una dirección

de hipercubo. Este pedido será escuchado por los posibles vecinos del nuevo nodo los cuales le ofrecerán una dirección que difiere en sólo una coordenada (un bit) respecto de la propia. Cada nodo de la red contará con, además de una dirección \mathcal{R} , una máscara que permitirá definir un espacio de direcciones que ese nodo administra. Tomando como ejemplo un hipercubo de dimensión $d = 4$, el nodo A con dirección y máscara 1000/3, estará cargado de administrar las direcciones 1000 y 1001. Cuando un nuevo nodo desee conectarse a la red, A podrá cederle la dirección 1001/4. En caso de que esto suceda, es decir que efectivamente el nuevo nodo acepte esta propuesta como su dirección \mathcal{R} , A aumentará su máscara a un valor de 4. Es en esta forma que A delega parte de su espacio de direcciones con su correspondiente administración a un nuevo nodo que se conecte a la red.

II-C. Ruteo reactivo

El direccionamiento basado en un hipercubo, presenta un aporte que simplifica el proceso de ruteo. Esto puede verse claramente si se considera el caso de un hipercubo completo, es decir aquel en el que todas las direcciones están siendo usadas por nodos. Para ilustrar esto último se puede considerar (ver 2) el caso en que el nodo 0001 quiere enviar un mensaje al nodo 1101 en un hipercubo de $d = 4$. Es posible ver la distancia entre ambos, que resulta ser de 2, por ser la diferencia en bits entre ambas direcciones. Al existir todos los nodos en la red, es posible saber a priori que el camino $0001 \rightarrow 1001 \rightarrow 1101$ será válido para rutear el paquete, y la cantidad de saltos que el mismo transitará es igual a la distancia entre los nodos. Como puede verse, el nodo origen sin disponer aún de ninguna ruta, ya sabe un posible camino al destino con distancia mínima. En un hipercubo incompleto, en cambio, algunas direcciones no estarán siendo efectivamente utilizadas por nodos, con lo cual no es posible asegurar un camino a priori. Sin embargo, es posible que algunos caminos con distancia mínima si existan y por lo tanto vale la pena intentar rutear el paquete en esa dirección. Esto mismo es lo que hace ANTop, ya que bajo su esquema de ruteo reactivo, los paquetes se envían en primera instancia a aquel vecino cuya distancia al destino sea menor, al tiempo que este último repetirá el procedimiento para lograr rutear el paquete. Si durante esta exploración, se llega a un callejón sin salida, el paquete será devuelto por donde vino para que el nodo que originalmente envió el paquete en esa dirección pueda enviarlo a su próximo vecino, eligiéndolo por el mismo criterio de distancia al destino.

II-D. Ruteo proactivo

El ruteo proactivo consiste de instalar en cada nodo rutas estables que básicamente se resuelve haciendo uso de la información ya existente en los nodos debido al mecanismo de asignación de direcciones tanto primarias como secundarias (II-F), y en base a determinados criterios que son detallados en el trabajo [11], permiten que diferentes nodos compartan información de ruteo y la distribuyan en un área de alcance acotado. Finalmente una vez que un nodo recibe información de ruteo de un vecino dependerá de los criterios de redistribución si la misma es propagada hacia

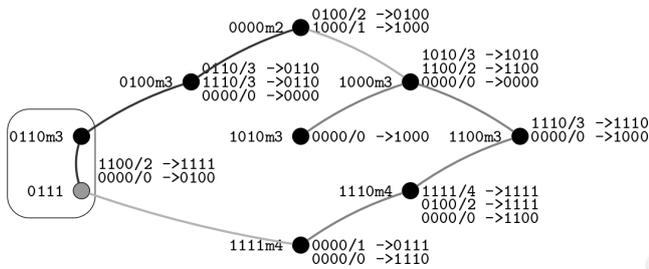


Figura 3. Ruteo proactivo.

otros vecinos cada vez más lejanos al nodo que inicialmente distribuye la información, luego será responsabilidad de cada nodo decidir si la información recibida le es de utilidad o no. En la figura 3 se muestra un ejemplo de una red junto con las respectivas tablas de ruteo en cada nodo, en la misma puede verse la estructura del árbol binario y las rutas por defecto que se instalan hacia los predecesores de primer orden y sucesores de primer orden.

II-E. Conexión de nodos en ANTop

En el protocolo ANTop no hay ningún servidor central ni jerarquías, por lo tanto todos los nodos tienen la misma funcionalidad. Lo primero que necesita un nodo para conectarse es obtener una dirección primaria, ya que de otra forma no podría ser distinguido de otros nodos. Una vez obtenida, puede comenzar a enviar y recibir paquetes de otros nodos, así como a ceder su espacio de direcciones o adquirir direcciones secundarias para mejorar la conectividad.

II-E0a. Establecimiento de conexión: Cuando un nodo desea conectarse a la red primero tiene que obtener una dirección primaria. Para ello, debe notificar a sus vecinos para que éstos le ofrezcan direcciones primarias. Esto lo hace enviando un paquete de tipo PAR (Primary Address Request) en modo broadcast, para que todos sus vecinos lo reciban y respondan a su vez con un paquete PAP (Primary Address Proposal), sugiriendo direcciones primarias. El primer nodo que se conecta a la red (es decir que no existe ningún otro nodo en su área de cobertura), no recibirá respuesta a cambio del envío del paquete PAR. Por esta razón el mismo asignará la primera dirección del hipercubo cuyos bits tiene un valor cero en su totalidad. El segundo nodo que quiera conectarse a la red, enviará un paquete PAR, en modo broadcast, y cuando el primer nodo participante de la red, recibe dicho pedido, debe responder con un paquete PAP, cuyo destino será la dirección del segundo nodo, es decir que no será enviado en modo broadcast. Luego de enviar el PAR, el nodo 2 espera un determinado tiempo para coleccionar las ofertas de dirección \mathcal{R} y vencido dicho intervalo, elige la opción con mayor máscara. Es decir, el espacio de direcciones más grande. En este caso, solo recibirá una respuesta del nodo 1. Una vez que el segundo nodo acepta la dirección, lo anuncia enviando un paquete PAN (Primary Address Notification) al nodo 1, notificando que acepta su propuesta y que éste incrementa su máscara a un valor de uno. Notar que el mensaje PAN se envía en modo broadcast para que todos los nodos que propusieron una dirección sepan que una fue elegida. El nodo que ofreció la dirección elegida cedería ese espacio de direcciones, y se enviará un paquete de tipo

PANC (Primary Address Notification Confirmation). Si el paquete PAN que envía el nodo que quiere conectarse a la red, se perdiese, entonces el nodo que ofreció la dirección elegida no enviará el PANC, y por lo tanto el proceso de conexión del nuevo nodo queda incompleto. Frente a este escenario, el nodo entrante vuelve a iniciar el ciclo de envío de paquetes PAR.

II-E0b. Dirección estable: Una vez que el proceso de conexión de un nodo a finalizado, el mismo tiene una dirección \mathcal{R} en el hipercubo y será vecino de todos aquellos nodos que cumplan con dos condiciones:

- Estar adentro del alcance de la placa inalámbrica
- Diferencia de un bit en las direcciones \mathcal{R} .

Al momento de rutear un paquete, cada nodo debe tener conocimiento de cuales de sus vecinos están activos en la red, de lo contrario, se perderán paquetes por ser reenviados a nodos que ya no forman parte de la red. Para lograr esto, cada nodo envía periódicamente un paquete de control del tipo HB (Heart Beat) en modo broadcast. Cuando un nodo recibe un HB, chequean en su tabla de vecinos si ya tiene registrado el origen como un vecino. En caso negativo, agrega una entrada al final de la tabla (generada por una lista de registros), en caso positivo se marca mediante una bandera que un nuevo HB ha sido recibido para ese nodo. Periódicamente, se recorre la tabla de vecinos y se chequean cuales vecinos no han enviado HB. En esos casos, se da un valor de uno a una variable que representa la cantidad de veces que no se ha recibido este aviso de actividad. Se repite este mecanismo, y se incrementa la variable si corresponde. Cuando llega un HB de ese vecino, la variable toma un valor nulo nuevamente. Cuando la misma llega un determinado valor máximo, configurable, se considera que el nodo ya no está en la red.

II-F. Establecimiento de dirección secundaria

Dado que este tipo de direcciones tienen el objetivo de mejorar la conectividad de los nodos, solamente se realizan una vez finalizado el proceso de dirección primaria. Además, dicha dirección está supeditada a la primaria. Dado que los nodos conectados se envían periódicamente paquetes HB, cada nodo al recibirlo puede comprobar si hay alguna dirección secundaria para asignarse y obtener conectividad con el vecino que envió el paquete. Si es posible asignar una dirección, entonces se envía un paquete SAP (Secondary Address Proposal), y se espera un paquete SAN (Secondary Address Notification) en respuesta. El paquete SAP contiene la dirección secundaria con el cual el nodo que lo reciba, tendrá conectividad con el emisor del paquete. En el caso que el vecino responda con el paquete SAN, el emisor del paquete SAP se auto asigna la dirección secundaria creando un enlace secundario. Luego de haber enviado un paquete SAP como propuesta de dirección secundaria, se espera un temporizador en el cual no se enviará paquetes SAP como propuesta a otros nodos, si no que se espera la confirmación SAN del vecino con el que se está negociando un enlace secundario. El nodo que recibe el paquete SAP, aparte de reenviar el mensaje de confirmación SAN, agrega como vecino la dirección secundaria propuesta y lo marca con conectividad por enlace secundario.

II-G. Desconexión de nodos

Cuando un nodo se desconecta, sus vecinos dejarán de recibir sus HB. Al cabo de un determinado tiempo, ellos consideran que el mismo abandonó la red, y lo eliminara de su tabla de vecinos. Quien sea el padre¹ del nodo, recuperará el espacio de direcciones cedido cuando el mismo se conectó, de modo tal que puede ser utilizado nuevamente. Cuando un nuevo nodo intenta conectarse dicho espacio es el primero en ofrecerse. Una vez que todos los recuperados fueron cedidos, se continua con la sesión del resto de las direcciones.

III. NUMERACIÓN DE REDES EN ANTOP

Dado que en este trabajo analizamos los casos de fragmentación y mezcla de redes, es necesario que cada red tenga un número de identificación que nos permita distinguirlas (que fue identificado en [2]); este aspecto no estaba contemplado en la definición original [1]. Uno de los principales inconvenientes en la mezcla surge en base a las posibles replicas de direcciones de hipercubo de nodos que anteriormente pertenecían a diferentes redes antes de mezclarse. Por ejemplo, la red 1 compuesta por los nodos A y B, con dirección y máscara 0000/1 y 1000/1 respectivamente; y una red 2 compuesta sólo por el nodo C con dirección 0000/0. Si se supone que el nodo C entra en el alcance del B, entonces este último va tener dos nodos vecinos (A y C) con la misma dirección, destruyendo así el concepto de direccionamiento por adyacencia, en el que se basa ANTop. Así mismo las mezcla de dos redes distintas podrían tener varias direcciones repetidas.

Por lo tanto proponemos extender en direccionamiento de ANTop de forma que las redes que son creadas independientemente estén identificadas por un campo del protocolo número de red. Dicho campo depende directamente de parámetros propio del nodo que tiene la primera dirección del hipercubo. En el momento que un nuevo nodo se conecte a alguna red, este hereda el número de red correspondiente a la red. En la fragmentación y mezcla de las redes ANTop, el número de red y la dirección relativa de los nodos irán variando dependiendo de la red o las redes formadas resultantes.

III-A. Algoritmo de asignación de número de red

En primer lugar se discute el valor que tiene que tener el número de red de una red ANTop.

La principal característica que debe tener el número de red es ser único e irrepetible. Como se ha explicado anteriormente, un nodo aislado crea una nueva red con la dirección con todos ceros, pero ahora deberá generar un número de red que esté relacionado a características únicas de cada primer nodo; por ejemplo la dirección MAC del adaptador Wi-Fi 802.11. Proponemos como identificación para de red ANTop uno de los tres bytes del campo NIC de la dirección MAC (del cuarto a; sexto byte de la MAC). Todos los demás nodos que se conecten mantendrán dicha identificación de red. En nuestra implementación usamos el primer byte de la NIC (el cuarto de la MAC). El algoritmo 1 de asignación de número de red está acompañado del

¹nodo padre es aquel que cedió la dirección primaria al conectarse a la red.

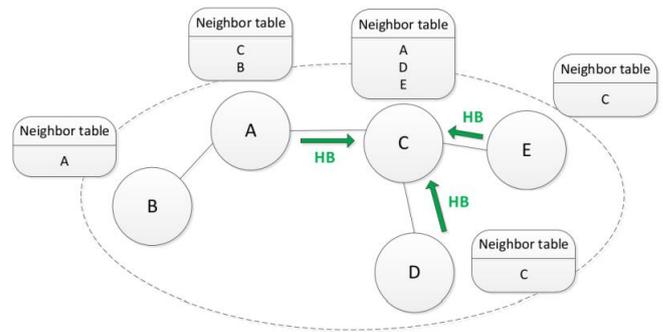


Figura 4. Red ANTop antes de la fragmentación.

algoritmo de asignación de dirección primaria. En decir, que acompaña los tiempos y procesos de asignación de la misma manera como se genera la dirección primaria.

Algorithm 1 Asignación de número de red

```

1: if se recibió al menos un paquete PAP con dirección disponible
   then
2:   Esperar a que llegue un paquete PANC o que transcurra
   un tiempo T_PANC;
3:   if se recibió un paquete PANC then
4:     Se asigna al número de red el valor del campo de
   número de red del mensaje PAP con menor máscara;
5:   end if
6: else
7:   if se recibió un paquete PAR indicando que no hay direc-
   ciones disponibles then
8:     No se puede conectar y no se asigna ningún valor de
   número de red;
9:     Fin del algoritmo;
10:  end if
11:  Al número de red se asigna el valor del cuarto byte de la
   dirección MAC del dispositivo;
12: end if

```

IV. FRAGMENTACIÓN DE REDES ANTOP

En la figura 4 se muestra una red ANTop cuya estructura se presentan las tablas de vecinos de cada uno de los nodos. Notar que los nodos tienen registrados las direcciones \mathcal{R} de sus vecinos porque reciben los paquetes HB. Por ejemplo el nodo C recibe paquetes HB de los nodos A, D y E; y por lo tanto C tiene registrado las direcciones \mathcal{R} de cada uno de ellos en su tabla de vecinos. Ahora, si el nodo A se desplaza físicamente hacia la izquierda a tal punto de salir del alcance del nodo C, el nodo C en principio va mantener en su tabla de registros de vecino a la dirección \mathcal{R} del nodo A, como también el nodo A mantendrá a C en su tabla de vecinos. Luego de tres ciclos de envíos de HB, cuyo temporizador del ciclo es configurable en la implementación del protocolo, tanto el nodo A como el nodo C, se van a eliminar de sus tablas de vecinos respectivamente (ver figura 5). Se ve claro que la fragmentación ocurrió, y esquemáticamente se observa dos redes separadas como resultado. De las dos redes resultantes, una tiene que tomar el rol de disparar el mecanismo de fragmentación, y la otra directamente no tomará acción alguna.

Suponiendo que en la figura 5, el nodo A es el padre del nodo C, entonces la red fragmentada será aquella en el cual

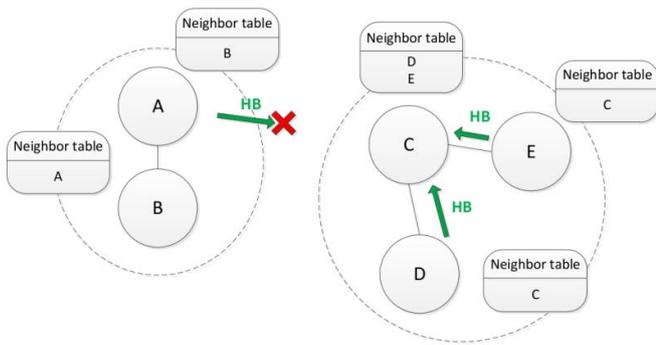


Figura 5. Red ANTop fragmentada.

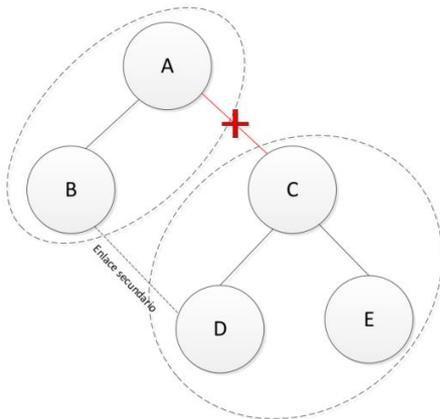


Figura 6. Fragmentación de red ANTop con enlace secundario.

está compuesta por los nodos C , D y E . En principio se ve que es necesario que un nodo pierda a su padre, pero no es condición suficiente.

Dado la existencia de las direcciones secundarias, el escenario de la figura 6 muestra un caso en el que no hay fragmentación. Por lo tanto, para verificar que no existan otras conexiones, hay que disparar un mecanismo de exploración para asegurar que no existan otros enlaces (direcciones secundarias).

El algoritmo 2 detector de fragmentación, es un proceso que corre cada vez que el nodo recibe un paquete HB.

IV-A. Proceso de asignación de direcciones

En principio una red fragmentada con las direcciones \mathcal{R} y máscara que fueron definida antes de la fragmentación, se podrían conservar y cumplir con las características de una red ANTop. Sin embargo la red fragmentada conserva el mismo número de red que la red original, y como se verá más adelante, este campo será importante que se diferencie entre redes independientes ante una posible futura mezcla de redes ANTop. Entonces es inevitable disparar un mecanismo que informe a todos los nodos de la red fragmentada para cambiar su número de red. El nodo que detecta la fragmentación, se asigna la primer dirección del hipercubo y elige su dirección de red (distinta a la actual), y dispara el mecanismo de renombramiento de los nodos en la red fragmentada.

En la figura 7 se muestra una red ANTop, conformada por los nodos A , B , C , D y E . En el momento que el

Algorithm 2 Detector de Fragmentación

Require: paquete HB

Ensure: None

- 1: **if** se recibió un paquete HB **then**
- 2: Se procesa el paquete HB;
- 3: Se actualiza la tabla de vecinos;
- 4: **goto** línea 7;
- 5: **else** return;
- 6: **end if**
- 7: **if** si hay un vecino que tiene menor distancia a la raíz que el padre actual **then**
- 8: Agrego al vecino como padre;
- 9: **end if**
- 10: **if** Si el padre no es un vecino **then**
- 11: **if** Si no se encuentran enlaces secundario entre las partes **then**
- 12: Iniciar proceso de fragmentación;
- 13: return;
- 14: **end if**
- 15: **else** return;
- 16: **end if**

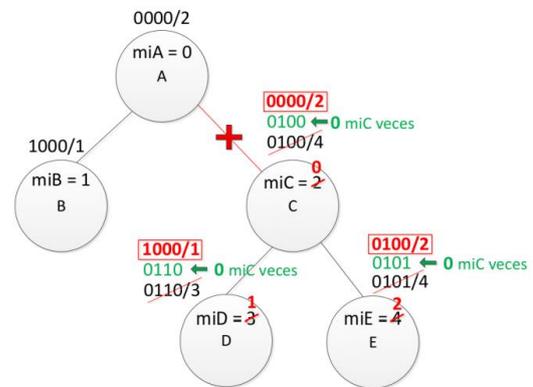


Figura 7. Direccionamiento de red ANTop fragmentada.

nodo C pierde conexión con el nodo A , los nodos C , D y E forman la red fragmentada. El nodo C es el que detectará la fragmentación y será el encargado de disparar el mecanismo de fragmentación: se asigna la primer dirección del espacio del hipercubo (la dirección \mathcal{R} 0000). El nodo C envía un mensaje a los nodos D y E para que se ajusten sus parámetros del protocolo y que se conserve la relación de vecinos entre sí. Lo que se hace es que los nodos subordinados desplacen los bits a izquierda de su dirección, agregando ceros desde la derecha. Esta última operación se la conoce como la operación de `shift` de ceros a izquierda. La cantidad de operaciones de desplazamiento dependerá de un valor en particular que se llama máscara inicial (mi)² del nodo que dispara el mecanismo de fragmentación (nodo detector).

IV-B. Paquetes de control

En el proceso de fragmentación, a partir del nodo detector y sus nodos subordinados recibirán mensaje de control que transmite la información necesaria para que puedan cambiar su dirección \mathcal{R} y máscara. Los paquetes de control que se utiliza en el proceso de fragmentación serán FAR (Fragment Address Request) y FAN (Fragment Address Notification).

² mi : valor de la máscara obtenida al conectarse a la red.

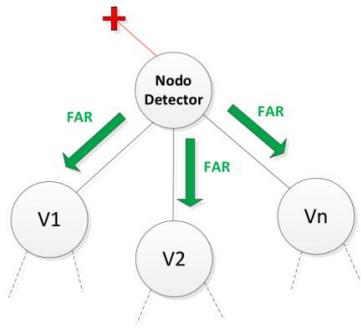


Figura 8. Envío de paquetes FAR desde el nodo detector.

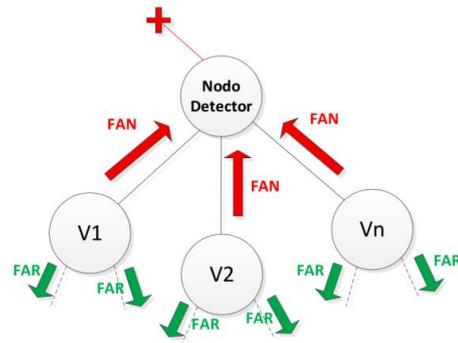


Figura 9. Envío de paquetes FAR y FAN de nodos intermedios.

El mensaje FAR básicamente transmite el valor de la máscara inicial del nodo detector, mientras que el FAN es un mensaje de confirmación del haber recibido correctamente dicho FAR. En el trabajo [13] se pueden consultar en detalle los campos y la información que portan.

IV-C. Algoritmos de envío de mensaje de control

Los algoritmos desarrollados son tres. El primero es ejecutado en el nodo detector en el momento que detecta la fragmentación, el segundo algoritmo se ejecuta cuando se recibe un paquete FAR, y el tercero cuando se recibe un mensaje FAN.

IV-C0a. Fragmentación: Los mensajes FAR y FAN, se deben enviar a todos los nodos de la red fragmentada, empezando desde el nodo detector hasta el final del árbol. El algoritmo 3 se ejecuta en el nodo detector al momento que detecta que la fragmentación ocurrió, donde el mismo debe avisar a todos sus nodos vecinos (hijos), barriendo su tabla de vecinos y enviando a cada uno de ellos un mensaje FAR como se muestra en la figura 8.

Algorithm 3 Fragmentación

- Require:** None
Ensure: None
- 1: Suspender el envío y procesamiento de paquetes HB;
 - 2: Suspender el servicio *Rendez Vous*;
 - 3: Asignar *número de red* (4to byte MAC del nodo detector);
 - 4: Shift de ceros a izquierda m_i veces a la dirección primaria;
 - 5: $m = m - m_i$;
 - 6: Ajustar la tabla de direcciones recuperadas;
 - 7: Borrar tabla de rutas;
 - 8: Enviar FAR a los hijos que tienen enlace principal;
 - 9: Borrar tabla de nombres *Rendez Vous*;
 - 10: Borrar tabla de vecinos;
 - 11: Setear la máscara inicial en cero;
 - 12: return;

IV-C0b. Procesamiento FAR: En el momento que un nodo recibe un paquete FAR, debe seguir con el envío hasta el final del árbol. Por ende, en el procesamiento del FAR, debe de enviar en principio el paquete FAR a todos los vecinos subordinados si es que tiene, y también el paquete de confirmación FAN al nodo del quien recibió el FAR, como se muestra en la figura 9. El algoritmo 4, muestra el procesamiento del FAR como también resuelve la dirección \mathcal{R} y máscara que se debe asignar cada nodo.

IV-C0c. Procesamiento FAN: El nodo que envía un mensaje FAR, debe de suspender el servicio de envío HB

Algorithm 4 Procesamiento FAR

- Require:** Paquete FAR
Ensure: None
- 1: Suspender el envío y procesamiento de paquetes HB;
 - 2: Suspender el servicio *Rendez Vous*;
 - 3: **if** la dirección destino del FAR es la dirección secundaria **then**
 - 4: Enviar confirmación FAN al nodo que envió FAR;
 - 5: Shift de ceros a izquierda m_{FAR} veces a la dir. secundaria;
 - 6: $m_{secundaria} = m_{secundaria} - m_{FAR}$;
 - 7: **if** el tratamiento de dirección primaria ocurrió **then**
 - 8: Reanudar servicio de HB y *Rendez Vous*;
 - 9: **end if**
 - 10: return;
 - 11: **end if**
 - 12: Agregar como vecino y padre al nodo que envió FAR;
 - 13: Enviar confirmación FAN al nodo que envió FAR;
 - 14: Asignar el *número de red* del campo del paquete FAR;
 - 15: Shift de ceros a izquierda m_{FAR} veces a la dir. primaria;
 - 16: Ajustar la tabla de direcciones recuperadas;
 - 17: Borrar tabla de rutas;
 - 18: Borrar tabla de nombres *Rendez Vous*;
 - 19: Enviar FAR a los hijos que tienen enlace principal;
 - 20: Borrar tabla de vecinos excepto el nodo padre;
 - 21: $m = m - m_{FAR}$;
 - 22: $m_i = m_i - m_{FAR}$;
 - 23: **if** no se envió paquete FAR **then**
 - 24: **if** no tiene dirección secundaria o el tratamiento de la dirección secundaria ocurrió **then**
 - 25: Reanudar servicio de HB y *Rendez Vous*;
 - 26: **end if**
 - 27: **end if**
 - 28: return;

y los paquetes de registro y resolución a los servidores *Rendez Vous*. La única manera de que se restablezca, es por medio de la recepción de un mensaje FAN de algún nodo subordinado. En el caso que un nodo contenga una dirección secundaria, deberá esperar a que termine el proceso de asignación de la nueva dirección \mathcal{R} y máscara, tanto la primaria como la secundaria, para restablecer dichos servicios. Esto último se muestra en el algoritmo 5 de procesamiento de FAN.

V. MEZCLA DE REDES ANTOP

En la siguiente figura 10, se encuentran dos redes ANTop independientes y numeradas. En cada nodo tiene identificada la dirección primaria, la máscara y el *número de red* que es igual a todos los nodos que pertenecen a la misma red. Por ejemplo, el nodo B tiene la dirección primaria 1000, la máscara 1 y el número de red 3d. Suponiendo que uno de

Algorithm 5 Procesamiento FAN

Require: Paquete FAN

Ensure: None

- 1: **if** si aún no se restableció el servicio de HB y *Rendez Vous* **then**
- 2: **if** no tiene dirección secundaria o el tratamiento de la dirección primaria y secundaria ocurrió **then**
- 3: Reanudar servicio de HB y *Rendez Vous*;
- 4: **end if**
- 5: **end if**
- 6: return;

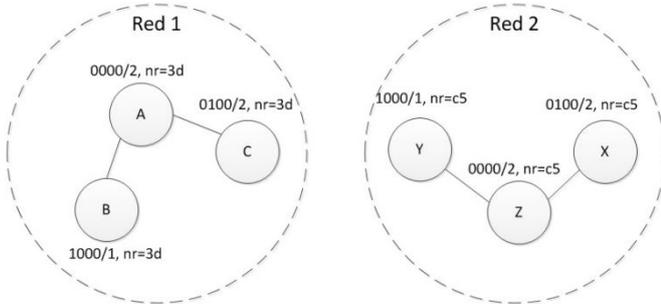


Figura 10. Dos redes ANTop independientes y numeradas.

los nodos se desplaza físicamente hasta lograr la mezcla de ambas redes. Si el nodo Y se desplaza sin perder vínculo de comunicación con el nodo Z y a su vez entra dentro del alcance del nodo C perteneciente a otra red ANTop, va a intercambiar mensajes de control ANTop pero ahora con un extra de información aportada por el campo *número de red* que nos permitirá saber a que red pertenece y armar una lógica de decisión según el caso que se presente.

En el momento que el nodo Y reciba el mensaje HB del C como se muestra en la figura 11 (y por lo tanto C cuando reciba el HB de Y), detecta que los números de red difieren, por lo que dispara el mecanismo de mezcla. Éste consistirá en el renombramiento de direcciones de una red ANTop a otra, y habrá que elegir cuál de los dos se elige.

La red que será renombrada en una mezcla es aquella que tiene el mayor *número de red* entre ambas (ver su justificación en el trabajo [13]). El algoritmo 6 muestra como decide una red en el momento de la mezcla con otra, y se inicia con la recepción del HB. El proceso de mezcla lo inicia la red que no será renombrada, ya que ésta tendrá que pasarle ciertos parámetros a la otra red (la que será renombrada). A su vez, dicho algoritmo sirve como filtro para procesar solo los paquetes HB que tienen un *número de red* igual a la red a la que pertenece.

V-A. Proceso de asignación de direcciones

Este proceso procura que se mantenga la estructura de direcciones \mathcal{R} basados en hipercubo, cuyo concepto es la esencia del protocolo ANTop y porque es más económico. En el esquema mostrado anteriormente de la figura 11, en el momento que se mezcla las dos redes ANTop por medio del par de nodos C-Y, el nodo C activa el mecanismo de mezcla renombrando a la Red 2 según la jerarquía de asignación de dirección del mismo a partir de su dirección \mathcal{R} 0100/2. El resultado de la mezcla se muestra en la figura 12, en donde

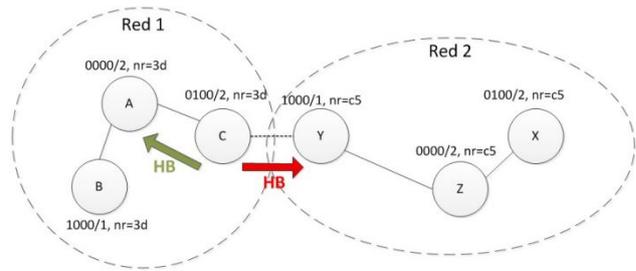


Figura 11. Mezcla de dos redes ANTop numeradas.

Algorithm 6 Detector de Mezcla

Require: Número de red del paquete HB

Ensure: None

- 1: **if** número de red local < número de red del paquete HB **then**
- 2: Iniciar proceso de mezcla;
- 3: No procesar paquete HB;
- 4: return;
- 5: **end if**
- 6: **if** número de red local = número de red del paquete HB **then**
- 7: Procesar paquete HB;
- 8: return;
- 9: **end if**
- 10: **if** número de red local > número de red del paquete HB **then**
- 11: No procesar paquete HB;
- 12: return;
- 13: **end if**

se agrego un bit más en las direcciones para mostrar en detalle las direcciones asignadas.

En la mezcla se muestran los cambios en color rojo, tanto de direcciones \mathcal{R} como también de máscara y *número de red* para los nodos que integraban la Red 2 y el nodo C en particular de la Red 1. Las máscaras cambian ya que deben respetar el espacio de direcciones asignados según el nodo que le antecede y le sucede. Por ejemplo, el nodo C al ceder el espacio de direcciones 01100 con máscara 2 a Y, debe aumentar en 1 su máscara dando como resultado máscara 3. Luego Y resulta de máscara 4 porque le cede el espacio de direcciones 01110 y máscara 4 al nodo Z, y se continúa por todos los nodos de la Red 2. El resultado es similar al proceso de asignación de dirección dado primero por la conexión del nodo Y enviando un paquete PAR en modo broadcast y el nodo C asignándole una dirección con un mensaje PAP. Luego el proceso es completado con los mensajes PAN y PANC. Se repite el mismo proceso cuando el segundo nodo en conectarse sea el Z que le llega una propuesta de dirección \mathcal{R} a partir de la recepción del mensaje PAP del nodo Y, y con el resto de los nodos. Sin embargo lo último descrito, sería como la interpretación del resultado similar a la del mecanismo de mezcla que se propone en esta sección. Es decir, que en la mezcla no habrá paquetes de control PAR, PAP, PAN o PANC para el renombramiento de una de las dos redes, sino que se diseñan nuevos paquetes de control que se definen más adelante. En el proceso de asignación de direcciones se dividen en dos vetas, es decir, que el mecanismo de mezcla en realidad estará compuesto por dos mecanismos distintos en el que a uno lo llamaremos mecanismo 1 y al otro mecanismo 2.

V-A0a. Mecanismo 1: Para explicar el mecanismo 1, se plantea los siguientes casos para renombrar una red

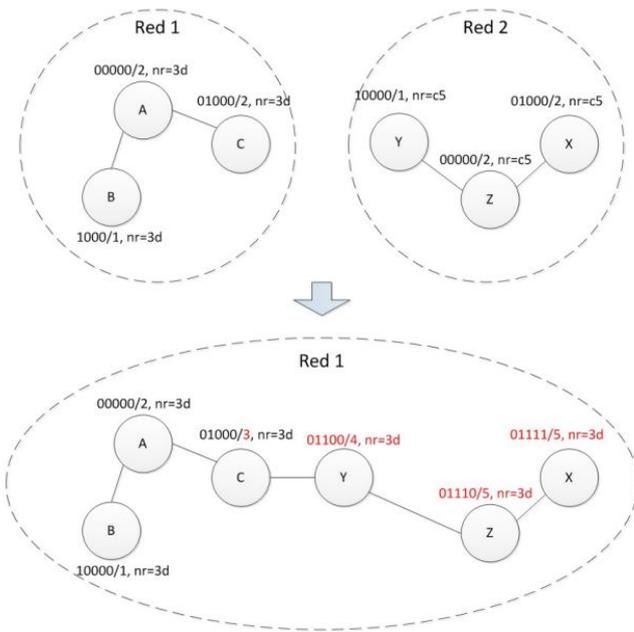


Figura 12. Resultado propuesto de la mezcla de dos redes ANTop.

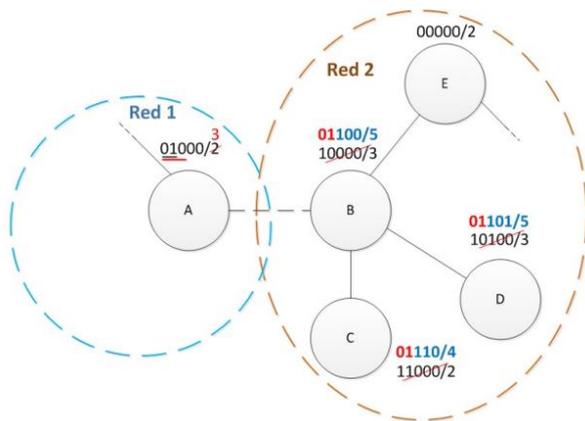


Figura 13. Primer caso de tratamiento de dirección del mecanismo 1 de mezcla.

ANTop en la mezcla.

Primer Caso. La idea es encontrar una solución para renombrar ciertos nodos de una red en el proceso de mezcla, sin que la topología y el orden de la estructura de direcciones \mathcal{R} formadas entre sí cambie lo menos posible. El primer caso se muestra en la figura 13.

En dicha figura, se observa que la mezcla comienza por medio del par de nodos A y B . Dado que la Red 2 tiene un identificador de mayor que la otra, el proceso de renombramiento se realiza en esta red.

En la topología del ejemplo planteado, como resultado de la nueva dirección del nodo B , está escrita en color rojo y azul arriba de la dirección 10000/3 antes de la mezcla. Se puede observar que está compuesta por dos cifras diferenciadas por los colores. El prefijo en rojo 01 coincide con el prefijo de la dirección 01000 con máscara 2 del nodo A . Luego el resto de la cifra en azul, coincide con la dirección del nodo mismo antes de la mezcla. Haciendo el mismo análisis para los nodos subordinado del nodo B ,

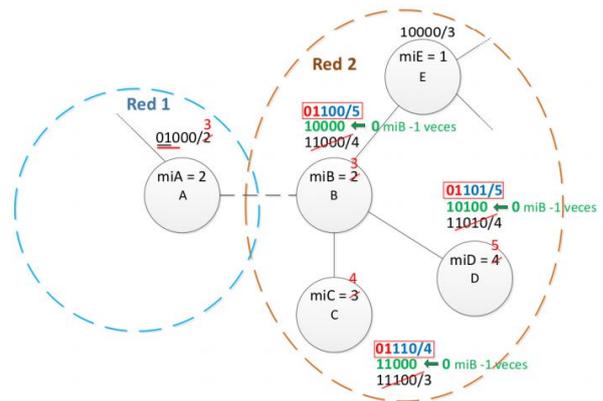


Figura 14. Segundo caso de tratamiento de dirección del mecanismo 1 de mezcla.

vemos que podemos aplicar el mismo criterio. Sólo el nodo E no cumpliría con la estrategia mencionada, ya que el resultado de la dirección \mathcal{R} luego de la mezcla sería 01000 que coincide con al dirección \mathcal{R} del nodo A . Entonces el criterio mencionado funciona para los nodos subordinado del nodo que forma parte del par que se establece la mezcla. Esta solución nos permite que las direcciones por los menos de los nodos subordinados permanezcan con la misma relación de distancias y jerarquía entre sí. Para el caso del nodo E , hay otro tratamiento de renombramiento de direcciones que se verá en el mecanismo 2. Las máscaras resultantes en los nodo B , C y D , se pueden obtener sumando su máscara de cada uno antes de la mezcla, con la máscara del nodo A antes de la mezcla (por ejemplo, inicialmente el nodo C tiene máscara 2). Luego de la mezcla, suma la máscara 2 del nodo A antes de la mezcla, obteniendo como resultado la máscara 4. Si se observa la máscara del nodo A después de la mezcla, resulta de ser 3 ya que en la mezcla toda la red quedará en el espacio de dirección dependiente del nodo C , por ende éste debe de ceder el espacio aumentando en uno su máscara.

Segundo caso. En el primer caso, el nodo E tiene la primera dirección del espacio de dirección por hipercubo, es decir, todos los bits significativos en cero. En el segundo caso el nodo E tiene un valor 10000/3 como se muestra en la figura 14.

Si aplicamos el mismo criterio del primer caso al nodo B , luego de la mezcla como resultado se obtiene la dirección $\mathcal{R} = 01110$. Calculando la distancia entre el nodo A y la dirección resultante del nodo B , obtenemos la distancia 2. La idea en la mezcla es que en el par de nodos que entran en juego y desencadenan la mezcla, estén a distancia 1, en donde no ocurre con el criterio del caso anterior. El mecanismo 1 en el esquema de la figura 14, solo será aplicado a los nodos B , C y D . Para que el nodo B sea un subordinado en dirección \mathcal{R} del nodo A , tiene que cumplir con dos características: que la resultante de la dirección \mathcal{R} tenga el prefijo de la dirección del nodo A hasta el valor de la máscara de este último, y que esté a distancia uno de dicho nodo A . Una de la maneras es que cada nodo realice un desplazamiento de ceros a la izquierda la cantidad de veces según el valor de la máscara inicial (mi) del nodo B . La cantidad exacta de dicha operación es del valor de

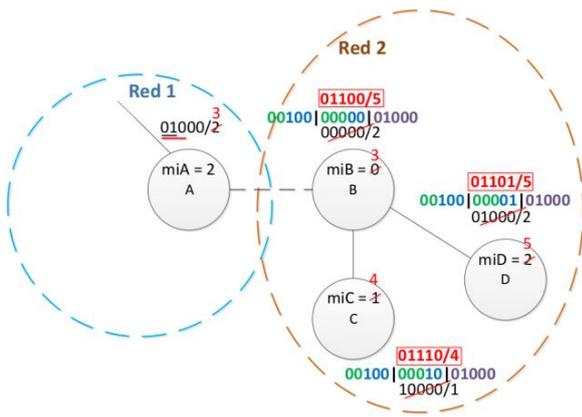


Figura 15. Tercer caso de tratamiento de dirección del mecanismo 1 de mezcla.

$mi(B) - 1$ del nodo B (ver el trabajo [13]). Por ejemplo, el nodo C, luego de la cantidad de $(2 - 1)$ operaciones de desplazamiento de ceros, se obtiene 11000; y aplicando el criterio del primer caso visto, obtenemos la dirección \mathcal{R} resultante 01110. Si se realiza el mismo procedimiento para el nodo B y D, se obtienen los mismos resultado que en el primer caso.

Luego se realiza el proceso visto en el caso anterior, en donde se vuelve a realizar el desplazamiento de ceros pero esta vez a la derecha y la cantidad del valor de la máscara del nodo A antes de la mezcla. Por último se realiza una operación OR con la dirección del nodo A. La máscara luego de la mezcla se podría obtener con la suma de la máscara antes de la mezcla, más la máscara del nodo A también antes de la mezcla. Pero ahora hay que considerar el concepto de la operación extra que trató en este caso. Cuando se menciono que para obtener la dirección \mathcal{R} resultante, en primer medida es necesario realizar el desplazamiento de ceros a izquierda la cantidad de $mi(B) - 1$ veces. En realidad lo que se está realizando es que se aumenta el espacio de direcciones la cantidad de $mi(B) - 1$, es decir, que la máscara resultante ahora hay que restarle $mi(B) - 1$ a la suma de la máscara local y máscara del nodo A antes de la mezcla. La máscara después de la mezcla se puede obtener aplicando la siguiente ecuación 1.

$$m_{mezcla} = m(A) + m - (mi(B) - 1) \quad (1)$$

La máscara inicial también de debe fijar nuevamente ya que si en un futuro vuelve a ocurrir una mezcla o fragmentación, puedan ejecutarse correctamente los algoritmos desarrollados en este artículo. La máscara inicial resultante se puede calcular con la siguiente ecuación 2.

$$mi_{mezcla(x)} = m(A) + 1 + (mi(x) - mi(B)) \quad (2)$$

Tercer caso. Falta un caso más por analizar, y es cuando el nodo B tiene la dirección \mathcal{R} con todos los bits en cero como se muestra en la figura 15. Todos los nodos de la Red 2 sin contar el B, serán subordinados de este último, por ende el tratamiento de direcciones que veremos será válida para todos los nodos de dicha red.

El criterio del primer caso no se podrá aplicar a este esquema, ya que el nodo B al realizar el desplazamiento

de ceros a derecha de la cantidad de máscara 2 del nodo A, obtendremos la misma dirección 00000. Y cuando se agregue el prefijo 01 del nodo A, se tendrá como resultado la misma dirección del nodo A. Como primera medida de debe crear una dimensión por medio del vector 10000. Dicho vector tiene la característica de poseer el bit más significativo en 1, y luego se podrá utilizar en cada nodo, desplazándolo a derecha y crear la dimensión faltante. La estrategia consistirá primero de tomar el vector 10000 y realizarle al operación de desplazamiento de ceros a derecha la cantidad de veces la máscara del nodo A. Luego se vuelve a realizar la operación de desplazamiento de ceros a derecha a la dirección \mathcal{R} actual de cada nodo, la cantidad de veces la máscara del nodo A más uno. Por último, se realiza la operación OR en entre las dos direcciones que obtuvimos más la dirección \mathcal{R} del nodo A. Por ejemplo, el nodo D se obtiene su dirección \mathcal{R} luego de la mezcla de la siguiente operación.

$$00100|00001|01000 \rightarrow 01101$$

1. 00100 : El vector 10000, se realiza la operación de desplazamiento de ceros hacia derecha la cantidad de veces la máscara 2 del nodo A.
2. 00001 : la dirección \mathcal{R} actual del nodo D 01000, se realiza la operación de desplazamiento de ceros la cantidad de veces la máscara 2 del nodo A más 1, o sea 3.
3. 01000 : Dirección \mathcal{R} del nodo A.

El último ejemplo se puede aplicar para todos los nodos de la Red 2, y se podrá observar que cumple con lo mostrado en la figura 15. Los valores de las máscara y mi de cada nodo, se podrá aplicar las mismas ecuaciones 1 y 2 del segundo caso.

V-A0b. Mecanismo 2: En la figura 16 del primer caso del mecanismo 1, el nodo E es antecesor del nodo B en donde se produce la mezcla. Como se comento, el mecanismo 1 contempla a aquellos nodos que son subordinados del nodo B, por ende, el nodo E queda fuera del alcance de dicho mecanismo. Por tal motivo se desarrolla el mecanismo 2, que se enfoca en resolver las direcciones \mathcal{R} de todos aquellos nodos que no son subordinados al par receptor en una mezcla de dos redes ANTop. En la figura 16, tenemos el caso de una mezcla entre el nodo A y B. Tanto el nodo B y C, corren el mecanismo 1 y se asignan las direcciones y máscara según en el primer caso. Para los nodos que pertenecen a la rama del nodo E desde el nodo B, tendrán un tratamiento particular en la dirección y máscara. En este caso el mecanismo 2 tendrá efecto sobre los nodos E y F.

Cuando el nodo B notifica a su vecinos sobre la existencia de la mezcla, los mensajes que le transmitirá a C será distinto el mensaje que le transmitirá a E. En primer lugar el nodo C necesita cierta información propia del mecanismo 1, en cambio el nodo E recibirá otro tipo de información acorde al mecanismo 2. En primer lugar vamos a describir el tratamiento de las direcciones sobre los nodos E y F, que tiene como resultado en la figura 16. El mecanismo 2, propone básicamente que los nodos de la rama se vayan asignando las direcciones del espacio del hipercubo, según el valor de la dirección \mathcal{R} y máscara del nodo predecesor de la rama. Por ejemplo, el nodo E recibe el mensaje del

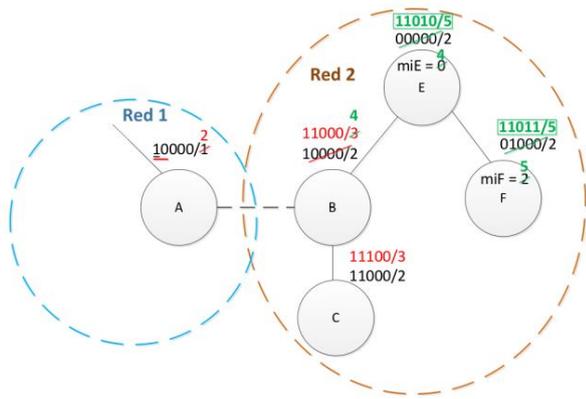


Figura 16. Tratamiento de direcciones mecanismo 2.

mecanismo 2 del nodo B , en donde le avisa que se asigne el próximo espacio de direcciones según la dirección y máscara actual del propio nodo B , es decir, dirección \mathcal{R} 11000 y máscara 3. Esto último, se considera que el nodo B ya se asignó su nueva dirección por medio del mecanismo 1 antes que le envíe el mensaje al nodo E por el mecanismo 2, en donde el resultado se ven en color rojo en el esquema, propios de los cambios del producidos por el mecanismo 1, y los cambios del mecanismo 2 se muestran en color verde. Entonces el nodo E , se asignará el próximo espacio dando como resultado la dirección \mathcal{R} 11010 y máscara 4. Luego el nodo B incrementa en 1 el valor de su máscara, dando como resultado 4. El mismo proceso ocurre entre el nodo E y F , en donde el nodo E envía un mensaje del mecanismo 2 al nodo F con la dirección 11010 y máscara 4. Este último se asigna la siguiente dirección del espacio de direcciones, o sea, dirección \mathcal{R} 11011 y máscara 5. Por último el nodo E incrementa en uno su máscara resultando el valor de 5. El valor de máscara inicial de cada nodo, se puede obtener con el valor de la máscara del nodo del que recibimos el mensaje por mecanismo 2, sumándole el valor de 1. Por ejemplo, el nodo E , toma el valor de la máscara 3 del nodo B y le suma 1, obteniendo una mi de 4.

V-A0c. Algoritmos de envío de mensajes de control: Los algoritmos desarrollados son cinco. En una mezcla, el nodo que comienza el proceso lo llamaremos par emisor, y el otro lo llamaremos par receptor (el de la red que cambia sus parámetro). Por ejemplo en la figura 12 el par emisor es el C , y el par receptor es el Y . Para el caso del mecanismo 1 los paquetes de control que se utiliza en el proceso de mezcla serán MAR1 (Mix Address Request 1) y MAN1 (Mix Address Notification 1). Para el caso del mecanismo 2 los paquetes de control que se utiliza en el proceso de mezcla serán MAR2 (Mix Address Request 2) y MAN2 (Mix Address Notification 2). El primero es el que será expuesto es el ejecutado en el par emisor. Luego se analizan los algoritmos que se ejecutan cuando se reciben los paquetes de control MAR1, MAN1, MAR2 y MAN2. **Mezcla.** En un proceso de mezcla, como se describió anteriormente, en principio entran en acción dos nodos en el que llamamos par emisor y par receptor. Cuando el par emisor detecta la mezcla por medio del nodo par receptor, este comienza el mecanismo de mezcla ejecutando el algoritmo 7 en el cual se encargará de enviar el mensaje MAR1 al

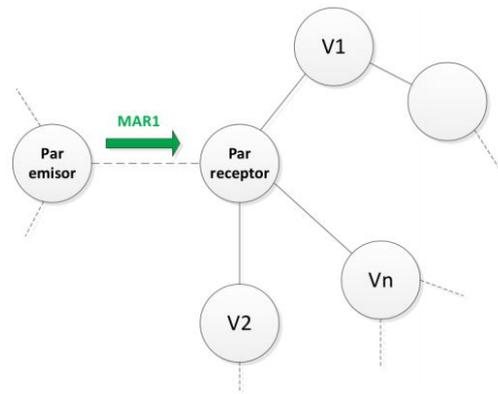


Figura 17. Envío del paquete MAR1 del par emisor al par receptor.

nodo receptor como se muestra en la figura 17.

Si el envío del mensaje MAR1 lo hace con su dirección \mathcal{R} , puede ocurrir que dicha dirección exista en la red con la que se mezcla, por ende antes de enviar el paquete MAR1, debe cambiar la dirección \mathcal{R} a una que llamaremos dirección por defecto. Esta última dirección tendrá como característica una dirección que no esté dentro del espacio de direcciones del hipercubo. Por otro lado, par emisor debe suspender los servicio de envío HB y los paquetes de registro y resolución a los servidores *Rendez Vous*.

Algorithm 7 Mezcla

Require: None

Ensure: None

- 1: Suspender el envío y procesamiento de paquetes HB;
- 2: Suspender el servicio de registración y resolución de nombres *Rendez Vous*;
- 3: Asignar como dirección \mathcal{R} , la *dirección por defecto*;
- 4: Enviar MAR1 al *nodo receptor*;
- 5: return;

Procesamiento MAR1. En la figura 17, muestra al nodo receptor recibiendo el MAR1 del nodo emisor, donde el nodo receptor dispara el mecanismo de envío de paquetes MAR1 a los nodos subordinados y MAR2 al nodo padre como se observa en la figura 18. En este caso los vecinos $V2$ y Vn recibirán el mensaje MAR1, y el $V1$ recibirá el mensaje MAR2. Antes se debe suspender el servicio de envío HB y el servicio de resolución y de registro a los servidores *Rendez Vous*. Luego del envío de los paquetes MAR1 y MAR2, se debe enviar el mensaje de confirmación MAN1 al par emisor. El algoritmo 8, describe el procesamiento de un paquete MAR1.

Si un nodo contiene dirección secundaria y no envió ningún mensaje MAR1 o MAR2, se deberá esperar que ambas direcciones, la primaria y la secundaria, se asignen las direcciones correspondiente en el proceso de mezcla para restablecer el servicio de HB y de registro y resolución de nombres *Rendez Vous*. También se muestra el calculo de máscara y mi según el caso de que se trate, si es par receptor o un nodo intermedio como lo son $V2$ y Vn en la figura 18. El nodo al correr dicho algoritmo distingue si tiene el perfil de par receptor o no. La distinción se debe por un lado a los diferencia del calculo de máscara y mi , y por el otro la necesidad de enviar un paquete MAR2 en el caso del par

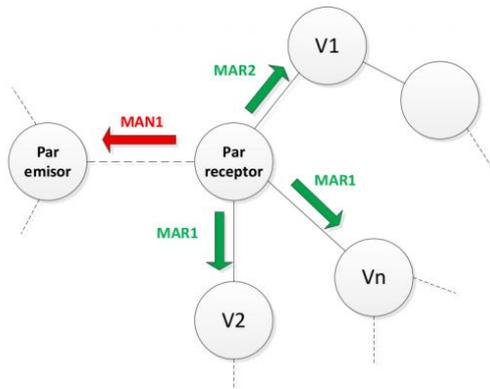


Figura 18. Envíos de MAR1 y MAR2 del par receptor, y confirmación MAN1 al nodo emisor.

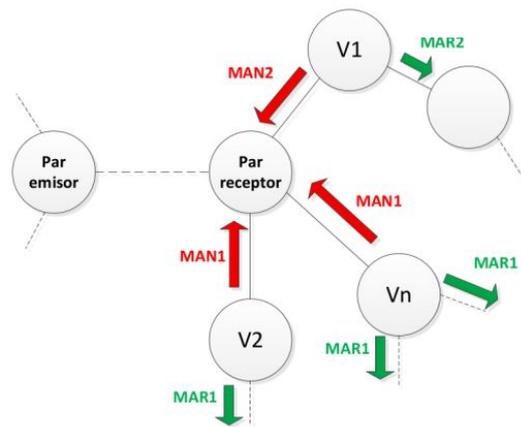


Figura 19. Propagación de mensaje MAR1, MAR2 y sus confirmaciones.

receptor, que en los demás nodos no es necesario.

Algorithm 8 Procesamiento MAR1

```

Require: Paquete MAR1
Ensure: None
1: Suspender el envío y procesamiento de paquetes HB;
2: Suspender el servicio Rendez Vous;
3: if la dirección destino del MAR1 es la dir. secundaria then
4:    $m_{sec} = m_{sec} + m_{paremisor} - (mi_{parreceptor} - 1)$ ;
5:   Enviar confirmación MAN1 al nodo que envió MAR1;
6:   Tratamiento de mecanismo 1 a la dirección secundaria;
7:   if el tratamiento de dirección primaria ocurrió then
8:     Reanudar servicio de HB y Rendez Vous;
9:   end if
10:  return;
11: end if
12: Asignar el número de red del campo del paquete MAR1;
13: if es el nodo receptor then
14:    $m = m + m_{paremisor} - (mi - 1)$ ;
15:    $mi = m_{paremisor} + 1$ ;
16: else
17:    $m = m + m_{paremisor} - (mi_{parreceptor} - 1)$ ;
18:    $mi = m_{paremisor} + 1 + (mi - mi_{parreceptor})$ ;
19: end if
20: Borrar tabla de nombres Rendez Vous;
21: if tiene vecinos a distancia 1 de la dirección primaria then
22:   Enviar MAR1 a los hijos que tienen enlace principal;
23:   Si es el par receptor y tiene nodo padre, enviarle MAR2;
24: end if
25: Borrar tabla de vecinos;
26: if no se envió paquete MAR1 o MAR2 then
27:   Enviar confirmación MAN1 al nodo que envió MAR1;
28:   Tratamiento de mecanismo 1 a la dirección primaria, y a la tabla de direcciones recuperadas;
29:   if no tiene dirección secundaria o el tratamiento de la dirección secundaria ocurrió then
30:     Reanudar servicio de HB y Rendez Vous;
31:   end if
32: else
33:   Enviar confirmación MAN1 al nodo que envió MAR1;
34: end if
35: return;

```

Procesamiento MAR2. En la figura 18, se muestra a V1 que recibe un paquete MAR2 del par receptor. Al procesar dicho paquete, V1 debe de seguir propagando el mecanismo 2 a todo los nodos de su rama, ver figura 19. Para ello primero se suspende el servicio de HB y de registro y resolución de nombres *Rendez Vous*. También se debe distinguir si el destino del paquete MAR2 recibido corresponde a la

dirección primaria o secundaria si es que tiene. En el caso de que el destino del mensaje MAR2 es la dirección primaria, luego se envía paquete MAR2 a todos los vecino que están distancia uno de la dirección primaria excepto al nodo del quien recibió el MAR2. Por último, para reanudar los servicios de HB y de registro y resolución de nombres *Rendez Vous*, se debe esperar hasta que tanto la dirección primaria como la secundaria se hayan tratados por los mecanismo correspondientes. En el caso de que no se haya enviado ningún paquete MAR2, ya que el mismo es el último de la rama el mecanismo 2, debe de asignarse la dirección \mathcal{R} correspondiente al mecanismo 2.

Procesamiento MAN1. En la figura 18, se muestra al nodo receptor enviando el mensaje de confirmación MAN1 al par emisor. Este al recibirlo y procesarlo, habilita a reanudar el servicio de HB y de registro y resolución de nombres *Rendez Vous* y también da la orden de volver a asignarse la dirección que tenía inicialmente antes de la mezcla. El mensaje de confirmación, avisa al emisor de paquetes MAR1, que recibió dicho paquete y por ende puede restablecer sus servicios y asignarse la dirección \mathcal{R} y mi con la que contará de ahora en más luego del proceso de mezcla; ver algoritmo 10.

Procesamiento MAN2. El procesamiento MAN2 del algoritmo 11 es muy similar al del MAN1, sólo se diferencia que al restablecer los servicios y asignar la dirección \mathcal{R} , utiliza los campos de datos y procedimientos del mecanismo 2. Por ende, el algoritmo cuando reconstruye la dirección \mathcal{R} final, debe tener el mismo efecto que si se hubiese reconstruido la dirección \mathcal{R} mediante procesamiento MAN1.

VI. CONCLUSIONES

En este artículo, hemos presentado los algoritmos de fragmentación y mezcla que completan las funcionalidades de ANTop, permitiendo que le mismo funcione correctamente para redes ad-hoc o de sensores, donde además de proveer una infraestructura es necesario asegurar un mínimo de consumo de energía y una maximización de la capacidad de transmisión de información. Tal como se ha mencionado en la introducción., otros protocolos para redes ad-hoc (BATMAN ó BABEL por ejemplo) construyen y mantiene tablas de ruteo por cada destino, y para ello utilizan los mecanismos de inundación de mensajes que son

Algorithm 9 Procesamiento MAR2**Require:** Paquete MAR2**Ensure:** None

```

1: Suspender el envío y procesamiento de paquetes HB;
2: Suspender el servicio Rendez Vous;
3: if la dirección destino del MAR2 es la dir. secundaria then
4:    $m_{secundaria} = mascara\ MAR2$ ;
5:   Enviar confirmación MAN2 al nodo que envió MAR2;
6:   Tratamiento de mecanismo 2 a la dirección secundaria;
7:   if el tratamiento de dirección primaria ocurrió then
8:     Reanudar servicio de HB y Rendez Vous;
9:   end if
10:  return;
11: end if
12: Asignar el número de red del campo del paquete MAR2;
13:  $m = mascara\ MAR2$ ;
14:  $mi = m$ ;
15: Borrar tabla de nombres Rendez Vous;
16: if tiene vecinos a distancia 1 de la dirección primaria then
17:   Enviar MAR2 a los hijos que tienen enlace principal,
    excepto el nodo que envió MAR2;
18:   Por cada MAR2 enviado, sumar 1 a  $m$ ;
19: end if
20: Borrar tabla de vecinos;
21: if no se envió paquete MAR2 then
22:   Enviar confirmación MAN2 al nodo que envió MAR2;
23:   Tratamiento de mecanismo 2 a la dirección primaria;
24:   Borrar tabla de direcciones recuperadas;
25:   if no tiene dirección secundaria o el tratamiento de la
    dirección secundaria ocurrió then
26:     Reanudar servicio de HB y Rendez Vous;
27:   end if
28: else
29:   Enviar confirmación MAN2 al nodo que envió MAR2;
30: end if
31: return;

```

Algorithm 10 Procesamiento MAN1**Require:** Paquete MAN1**Ensure:** None

```

1: if si aún no se restableció el servicio de HB y Rendez Vous
   then
2:   if la dirección primaria es igual a la dirección por defecto
     then
3:     Asignar la dirección  $\mathcal{R}$  primaria inicial;
4:      $m = m + 1$ ;
5:     Reanudar servicio de HB y Rendez Vous;
6:   else
7:     Tratamiento de mecanismo 1 a la dirección primaria y
     a la tabla de direcciones recuperadas;
8:     if no tiene dirección secundaria o el tratamiento de la
     dirección primaria y secundaria ocurrió then
9:       Reanudar servicio de HB y Rendez Vous;
10:    end if
11:   end if
12: end if
13: return;

```

Algorithm 11 Procesamiento MAN2**Require:** Paquete MAN2**Ensure:** None

```

1: if si aún no se restableció el servicio de HB y Rendez Vous
   then
2:   Tratamiento de mecanismo 2 a la dirección primaria;
3:   Borrar tabla de direcciones recuperadas;
4:   if no tiene dirección secundaria o el tratamiento de la
     dirección primaria y secundaria ocurrió then
5:     Reanudar servicio de HB y Rendez Vous;
6:   end if
7: end if
8: return;

```

excesivamente costosos en términos de energía y capacidad del canal para cada nodo. Estos costos se ven reflejados sobre todo para redes de gran tamaño, que el protocolo ANTop ha demostrado un correcto funcionamiento tal como se ve en los trabajos [6] y [11].

Por último, hemos puesto a disposición del público una implementación para Linux basados en Debian, ver [12].

REFERENCIAS

- [1] J. Ignacio Alvarez-Hamelin, Aline C Viana, and Marcelo D De Amorim. Dht-based functionalities using hypercubes. In *Ad-Hoc Networking*, pages 157–176. Springer, 2006.
- [2] Matías Campolo. Estudio y análisis del funcionamiento de ANTop sobre IPv6. http://cnet.fi.uba.ar/matias.campolo/Tesis_Matias_Campolo.pdf, 2012.
- [3] J. Chroboczek. The Babel Routing Protocol. RFC 6126 (Experimental), April 2011. Updated by RFCs 7298, 7557.
- [4] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms second edition*. The Press MIT press Cambridge, 2001.
- [5] Elis Kulla, Makoto Ikeda, Leonard Barolli, and Rozeta Miho. Impact of source and destination movement on manet performance considering batman and aodv protocols. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, pages 94–101. IEEE, 2010.
- [6] Alejandro Marcu. Desarrollo y simulación de un protocolo para redes adhoc. http://cnet.fi.uba.ar/alejandro.marcu/Tesis_Alejandro_Marcu.pdf, 2007.
- [7] A Neumann, C Aichele, and M Lindner. Better approach to mobile ad-hoc networking (batman) draft-wunderlich-openmesh-manet-routing-00. [URL: http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00](http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00) [cited 4 February, 2014], 2008.
- [8] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.
- [9] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsvd) for mobile computers. In *ACM SIGCOMM computer communication review*, volume 24, pages 234–244. ACM, 1994.
- [10] D. Savage, J. Ng, S. Moore, D. Slice, P. Paluch, and R. White. Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP). RFC 7868 (Informational), May 2016.
- [11] Gastón Teja. Estudio y análisis de mecanismos orientados al robustecimiento de ANTop, utilizando ruteo proactivo. http://cnet.fi.uba.ar/gaston.teja/Tesis_Gaston_Teja.pdf, 2010.
- [12] Pablo Torrado and J. Ignacio Alvarez-Hamelin. ANTop (Adjacent Networks Topologies) ad-hoc routing. <https://github.com/CoNexDat/ANTop>, 2018.
- [13] Pablo Damián Torrado. Fragmentación y Mezcla de redes ad-hoc utilizando el protocolo ANTop sobre IPv6. http://cnet.fi.uba.ar/pablo_torrado/Tesis_Pablo_Torrado.pdf, 2018.