Capítulo 4 Monitoreo remoto de redes (RMON)

4.1 Introducción

El monitoreo remoto de redes es la especificación más importante que se puede agregar al conjunto SNMP, MIB-II. Esta especificación se encuentra descripta en varias RFCs, las de mayor importancia son las siguientes:

- RFC1757: Remote Network Monitoring Management Information Base.
- RFC2021: Remote Network Monitoring Management Information Base II.

RMON define una MIB que amplía las funcionalidades ofrecidas por MIB-II sin la necesidad de realizar cambios en el protocolo de gestión SNMP. Con la MIB RMON se puede obtener información sobre la red misma, y no sobre los dispositivos conectados a la red como es el caso de MIB-II.
El uso de RMON para la gestión de redes ofrece muchas ventajas, algunas de estas son:

- *Operación off-line*: aún si se ha producido un fallo en la conexión entre el monitor RMON y la estación de gestión de red, el monitor continuará recolectando información a cerca de la red (performance, tráfico, etc.). De esta manera, cuando se restaure la conexión con el gestor, el monitor enviará la información requerida por éste. Además, como el monitor recolecta la información estadística sin necesidad de que el gestor envíe constantemente pedidos, se logra reducir el tráfico de polling, disminuyendo el ancho de banda necesario para gestión en la red.
- *Detección y reporte de fallos*: El monitoreo preventivo ofrece la posibilidad de detectar una fallo antes de que éste ocurra. Esto se logra mediante el análisis de la performance y el tráfico de la red. El monitor RMON puede configurarse para que detecte estos casos automáticamente, y en caso de ocurrir esta condición, la registra y envía un mensaje de notificación al gestor de red.
- *Datos con valor agregado*: El monitor de red puede realizar un análisis de la información recolectada de la subred que gestiona. Por ejemplo, puede reconocer las estaciones que generan mayor cantidad de tráfico, o las que generan mayor cantidad de errores en la red. Esta función no podría ser ejercida por el mismo gestor de red, salvo que se encontrara dentro de la misma subred.
- *Múltiples Gestores de red*: En casos donde la red es muy amplia y consta de muchas subredes puede ser necesario el uso de más de un gestor de red. El monitor de red puede ser configurado para que funcione correctamente con múltiples gestores de red.

Nota: no todos los monitores de red soportan estas características, pero la especificación provee el soporte para el uso de las mismas.

En la especificación RMON se define el explorador RMON (RMON probe) que es todo sistema que implemente el uso de la MIB RMON. El explorador RMON consta de un agente RMON, que no difiere de las características de un agente SNMP, y de una entidad de procesos RMON (RMON probe process entity) que es la que provee la capacidad de lectura y escritura en la MIB RMON local.
El monitor RMON puede encontrarse en un dispositivo totalmente dedicado para este servicio, o ser un proceso que se ejecuta en un dispositivo de la red, como por ejemplo un Switch, consumiendo recursos de memoria y de procedimiento para la función de gestión.

4.2 Configuración de RMON

La MIB RMON se divide en nueve grupos, donde cada uno provee funciones específicas para la gestión de la red. Estos grupos pueden implementarse en su totalidad, o sólo un conjunto de éstos. Pero en el caso de implementar un grupo este debe implementarse en su totalidad. Cada grupo puede constar de una o más tablas de control y una o más tablas de datos. Generalmente las tablas de datos son solamente de lectura (el explorador RMON es el que se encarga de escribir los datos). Las tablas de control son de lectura-escritura y contienen los parámetros necesarios para configurar los datos que aparecerán en las tablas de datos. Estos parámetros se configuran desde el gestor de red añadiendo filas a la tabla o modificando una fila ya existente. Para modificar un parámetro de la tabla de control debe invalidarse la fila correspondiente y crearse una nueva fila con los valores requeridos. Al invalidarse la fila de la tabla de control se eliminan automáticamente la fila de la tabla de control y las filas asociadas en la tabla de datos.

4.3 Ejecución de comandos mediante la MIB RMON

El protocolo SNMP no ofrece la posibilidad de enviar un comando a un agente para que realice una acción determinada, la única posibilidad que ofrece es la lectura y escritura de objetos definidos dentro de una MIB. Sin embargo, pueden realizarse acciones en el agente mediante el comando Set de SNMP. Para esto se definen objetos que representan un comando en lugar de información. Al cambiar el valor del objeto se ejecuta una acción específica.
En la MIB RMON se definen varios objetos de este tipo. Estos objetos representan estados, y una acción se llevará a cabo cuando el gestor de red cambie el estado de uno de estos objetos.

4.4 Gestores múltiples

Cuando un explorador RMON es compartido puede ser accedido desde diferentes gestores de red y pueden ocasionarse las siguientes dificultades:

1. Si se realizan pedidos de varios gestores a la vez donde cada uno necesita cierta cantidad de recursos, y la suma de estos excede la capacidad del explorador RMON, éste no podrá entregar dichos recursos ocasionando que alguno de los pedidos no se lleven a cabo.
2. Si una estación de gestión retiene recursos del monitor por un período largo de tiempo, puede provocar que otra estación de trabajo no pueda acceder a los datos de gestión por falta de recursos.
3. Los recursos pueden estar asignados a un gestor que haya sufrido un fallo sin haber liberado estos recursos.

Para evitar este tipo de problemas la MIB RMON tiene para todas las tablas de control definidas una columna que identifica al propietario de la fila correspondiente. Este valor puede ser usado de las siguientes maneras:

1. Un gestor puede reconocer los recursos que posee y ya no necesita, y de esta forma puede liberarlos.
2. Un operador de red puede conocer qué gestor está utilizando determinados recursos o funciones y puede negociar por éstos para que sean liberados.
3. Un operador puede tener la autorización para liberar recursos que otros operadores hayan reservado.
4. Si un gestor ha sido reinicializado puede detectar los recursos que poseía anteriormente y liberar los que ya no necesitará.

Nota: La especificación de RMON sugiere que en la etiqueta de propietario se coloquen uno o más de los datos siguientes: dirección IP, nombre de la estación de gestión, lugar donde se haya la estación, nombre del administrador de red y el número telefónico.

La etiqueta de propietario no confiere ningún tipo de seguridad sobre los objetos asociados. Estos pueden ser leídos, modificados y borrados por cualquier gestor de red (en caso de que los objetos tenga permisos de lectura y escritura).
Para mejorar la eficiencia del sistema en el uso de los recursos, en casos de gestores múltiples, pueden compartirse los valores obtenidos por el explorador RMON. Para que pueda llevarse a cabo esta mejora es necesario que cuando un gestor requiere información, debe observar antes de solicitar los recursos necesarios por si otro gestor está pidiendo los mismos datos. Si es así, puede leer los datos de las filas de la tabla creadas por el otro gestor, sin gastar más recursos del explorador RMON. La desventaja de compartir recursos se da cuando el propietario de las filas de la tabla ya no necesita más la información y elimina las mismas, sin saber que estaban siendo utilizadas por otro gestor.

4.5 Gestión de tablas

En SNMPv1 no se especifica claramente los procedimientos para agregar y borrar filas de una tabla. Como en el caso de RMON esto es sumamente necesario, se han definido nuevas convenciones textuales (*textual conventions*), que no contradicen la especificación SNMPv1, y ayudan a definir en forma clara y ordenada la manera en que deben agregarse y borrarse las filas de las tablas.

4.5.1 Convenciones textuales

En la especificación RMON se han agregado dos nuevos tipos de datos:

- *OwnerString*: es el tipo de dato que se asocia al objeto que indica el propietario en la fila de una tabla. Este dato es un mnemónico de *DisplayString* que representa una cadena de octetos (entre 0 y 255 octetos).
- *EntryStatus*: es el tipo de dato que se asocia al objeto que representa el estado de la fila en una tabla. Este tipo de objeto puede tomar los valores: *valid*, *createRequest*, *underCreation*, *invalid*.

4.5.2 Formato general de una tabla

En la figura 2.1 se muestran una tabla de control y la tabla de datos asociada con todas las características mencionadas en los puntos anteriores.
En el ejemplo se muestra que los objetos de la columna *dataControlIndex* en la tabla de datos están asociados a la fila que tiene el mismo valor en la columna *controlIndex* de la tabla de control. Además, se puede ver como se define el propietario de cada grupo de filas de datos mediante la columna *controlOwner* en la tabla de control.
Para agregar una fila en una tabla, el gestor debe enviar una SetRequest PDU con los identificadores de objeto de las columnas y sus valores correspondientes. Idealmente, deberían enviarse los valores de todas las columnas, pero no es necesario. Para evitar conflictos, si dos o más estaciones de gestión intentan crear una fila con los mismos parámetros (incluso el valor de el o los índices de la tabla) se realiza un intercambio de paquetes entre el agente y la estación de gestión. Este intercambio de paquetes es conocido como "RMON Polka". Los pasos a seguir para la creación de una fila son los siguientes:

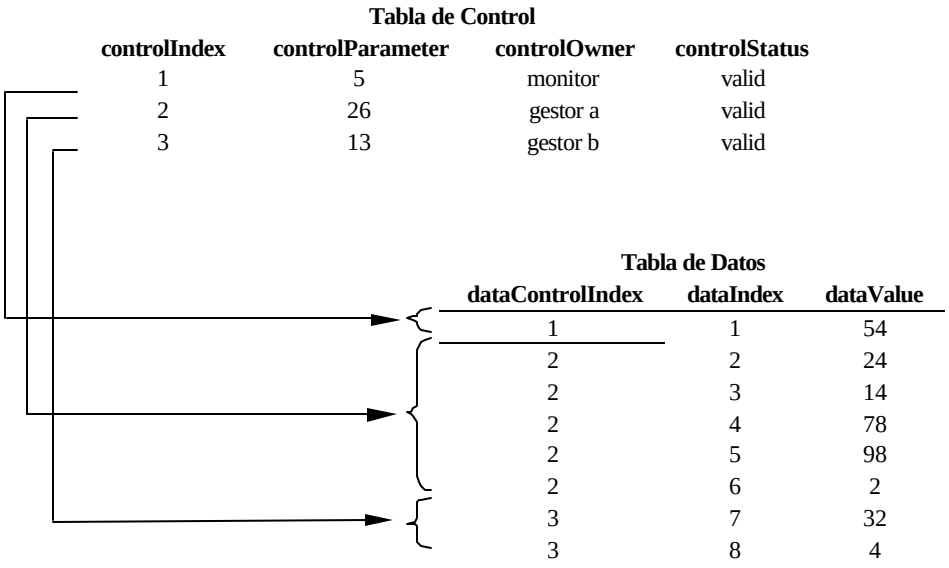

Figura 4.1: Ejemplo de tabla RMON.

1. Si una estación de gestión intenta crear una nueva fila, y el valor de cada uno de los índices aún no existe, la fila es creada con el valor del objeto *status* en *createRequest.*
2. Inmediatamente después de terminada la operación de creación de la fila, el agente cambia el valor del objeto *status* a *underCreation*.
3. La fila debe permanecer en el estado *underCreation* hasta que la estación de gestión termine de crear todas las filas que necesite para la configuración. Una vez terminada la configuración, la estación de gestión cambia el valor de *status* a *valid*.
4. Si se intenta crear una nueva fila, con el estado *createRequest*, y la fila ya existe, el agente devolverá un error.

Para borrar una fila de una tabla debe cambiarse el valor de *status* a *invalid* mediante la PDU SetRequest correspondiente. Mientras que para modificarla, debe cambiarse el estado a *invalid* y luego cambiar los valores deseados de la fila.
La figura 2.2 muestra las formas posibles en que puede cambiarse el estado de una fila.

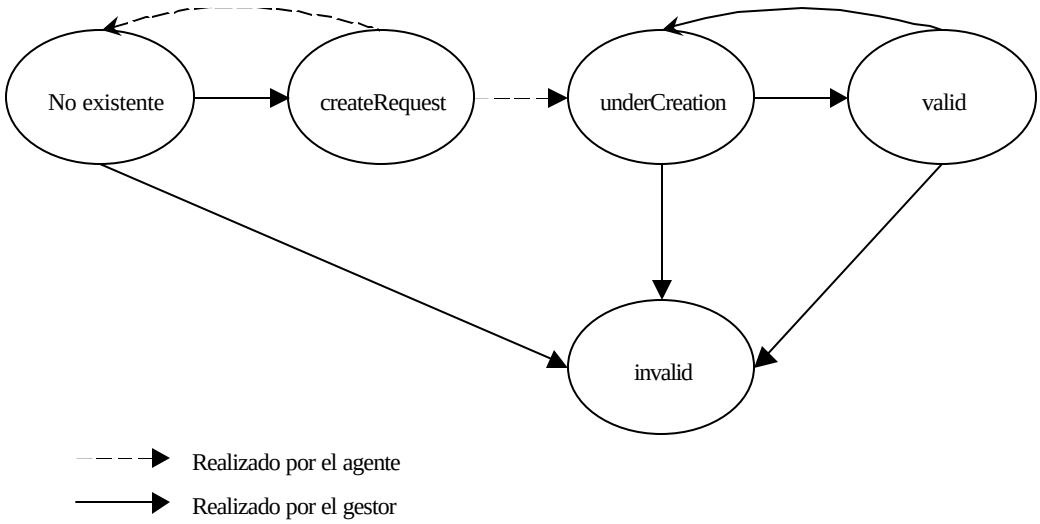

Figura 4.2: Transiciones de la variable **status**.

4.6 MIB RMON

El monitoreo remoto de redes está especificado con la definición de una MIB que se encuentra detallada en la RFC 1757. Esta MIB se encuentra incorporada en el árbol de MIB-II con el identificador 16. La MIB RMON está compuesta por nueve grupos:

1. statistics
2. history
3. alarms
4. host
5. hostTopN
6. matrix
7. filter
8. capture
9. event

En la RFC 1513 se definen extensiones en la MIB RMON para la gestión de redes tipo Token Ring (802.5). Esta extensión consiste en el agregado de nuevas tablas en los grupos ya existentes, *statistics* y *history*, y el agregado de un nuevo grupo:

10. tokenRing

Los diez grupos son opcionales para la implementación de RMON. Sin embargo, existen algunas dependencias que se detallan a continuación:

1. El grupo *alarm* necesita la implementación del grupo *event*.
2. El grupo *hostTopN* necesita la implementación del grupo *host.*
3. El grupo packet *capture* necesita la implementación del grupo *filter.*

A continuación se detallarán las funciones de cada uno de los grupos.

4.6.1 Grupo statistics

El grupo statistics almacena datos estadísticos sobre cada una de las subredes monitoreadas. En la RFC 1757 se define una única tabla para este grupo *etherStatsTable*. Cada una de las filas de la tabla representa una interfase de red Ethernet.
La tabla recolecta información sobre cada subred, incluyendo: cantidad de bytes, de paquetes, paquetes de determinada longitud, paquetes erróneos, cantidad estimada de colisiones, cantidad de paquetes broadcast y multicast. Esta información se almacena en objetos cuyo tipo es contador, que se inicializan en cero cuando es creada una nueva fila.
Además de los contadores la tabla está compuesta por otros cuatro objetos:

etherStatsIndex (entero): identifica cada una de las filas creadas con un entero.
etherStatsDataSource (String)*: identifica la interfase a la que hace referencia la fila, el valor de este objeto es el identificador de objeto de la instancia *ifIndex* en el grupo *interfaces* de MIB-II.
etherStatsOwner (OwnerString)*: identifica a la estación de gestión que creó la fila en la tabla.
etherStatsStatus (EntryStatus)*: identifica el estado de la fila (su uso fue explicado en párrafos anteriores).

* Estos son los únicos objetos de la tabla que tienen permiso de lectura y escritura (los demás son sólo de lectura).

El grupo *statistics* recolecta información acerca de la carga de la red Ethernet e información acerca de los fallos producidos en la misma, ya que detecta diversos tipos de errores, como por ejemplo:

error de alineación CRC, colisiones y paquetes con tamaño incorrecto (para Ethernet el tamaño de los paquetes puede ubicarse entre los 64 y los 1518 octetos).
La RFC 1513 define para este grupo dos nuevas tablas para redes TokenRing: la tabla de estadísticas de capa MAC (*tokenRingMLStatsTable*) y la tabla de estadísticas en modo promiscuo (*tokenRingPStatsTable*).
La primera de las tablas está constituida por contadores de paquetes de control de la capa MAC, incluyendo reportes de error.
La segunda de las tablas está constituida por estadísticas de todos los paquetes de datos enviados a través de la red Token Ring. Esta es la misma técnica usada en la tabla *etherStatsTable*.

4.6.2 Grupo history

El grupo *history* se encarga de almacenar muestras de algunas de las variables de las 3 tablas del grupo *statistics*.
Este grupo está conformado por 4 tablas:

- *historyControlTable*
- *etherHistoryTable*
- *tokenRingMLHistoryTable*
- *tokenRingPHistoryTable*

Nota: Las dos últimas tablas están definidas en la RFC 1513.

Cuando el gestor necesita obtener información acerca de la evolución de algunos parámetros de la red a través del tiempo, debe crear una nueva fila en la tabla de control incluyendo los datos necesarios:

- Interfase de la que se quieren obtener los datos.
- Cantidad de muestras a almacenar.
- Intervalo entre muestras.

Una vez creada la fila nueva en la tabla de control, el monitor RMON creará una fila nueva en la tabla de datos correspondiente a la interfase (Ethernet o TokenRing) por cada muestra tomada.

Al poder definirse el intervalo entre muestras puede obtenerse diferente información acerca de una misma red, cuanto más corto sea el intervalo el gestor podrá detectar cambios repentinos en la red, mientras que cuanto más largo sea el intervalo el gestor tendrá información acerca del estado "estacionario" de la red.

4.6.3 Grupo alarm

El grupo alarm define un conjunto de umbrales para monitorear la performance de la red. Si un umbral es cruzado en la dirección apropiada se generará un evento (definido en el grupo *event*) asociado a la alarma.
Este grupo consiste en una única tabla (*alarmTable*). Cada fila especifica la variable en particular que debe ser monitoreada, el intervalo entre muestras, los umbrales definidos para esta variable, y determina si se mide el valor de la variable o la variación respecto a la medición anterior (delta). Se debe notar que aunque se mencionen muestras, sólo se almacena la muestra más reciente y no un conjunto de muestras tal como lo hace el grupo *history*.

4.6.3.1 Funcionamiento de las alarmas

El grupo alarm define un mecanismo diseñado para prevenir que se generen alarmas repetidamente, en caso de que el valor de la variable monitoreada oscile en torno al valor del umbral. Las reglas que sigue este mecanismo son las siguientes:

1. (a) Si la primera muestra luego de que la fila sea válida es menor al valor del umbral de subida, se generará una alarma de subida la primera vez que el valor de la variable supere dicho umbral.
   (b) Si la primera muestra luego de que la fila sea válida es mayor o igual al umbral de subida, y el valor del objeto *alarmStartupAlarm* es *risingAlarm*(1) o *risingOrFallingAlarm*(3), se generará una alarma de subida.
   (c) Si la primer muestra luego de que la fila sea válida es mayor o igual al umbral de subida, y el valor del objeto *alarmStartupAlarm* es *FallingAlarm*(2), se generará una alarma de subida la primera vez que el valor de la variable supere el umbral de subida luego de haber descendido este umbral.
2. Luego de que se ha generado una alarma de subida, no podrá generarse nuevamente hasta que la variable haya descendido hasta el nivel del umbral de bajada y vuelto a subir hasta el umbral de subida.

Nota: Las reglas para la generación de alarma de bajada son las opuestas a las enunciadas anteriormente.
En las especificaciones de RMON se denomina mecanismo de histéresis. En la figura 4.3 puede verse una representación gráfica de este mecanismo.

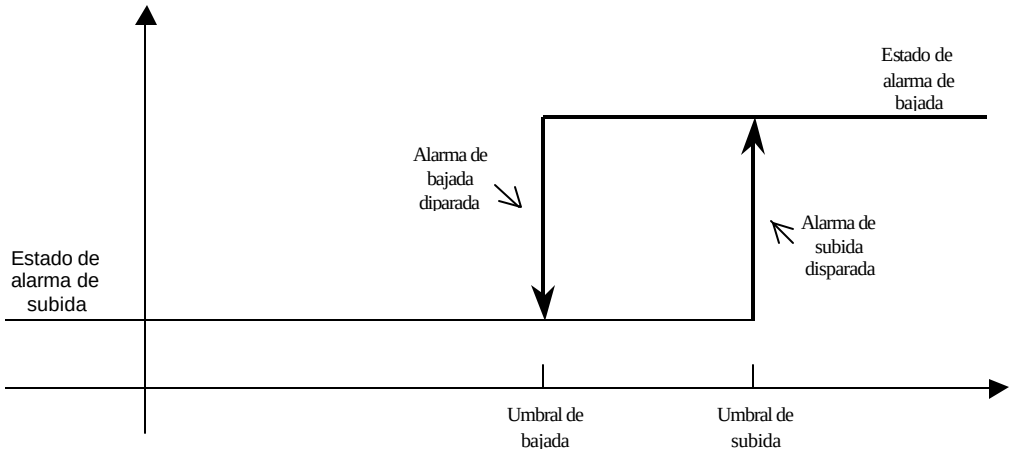

Figura 4.3: Mecanismo de histéresis

En la figura 4.4 mostramos con un ejemplo el funcionamiento de este mecanismo.

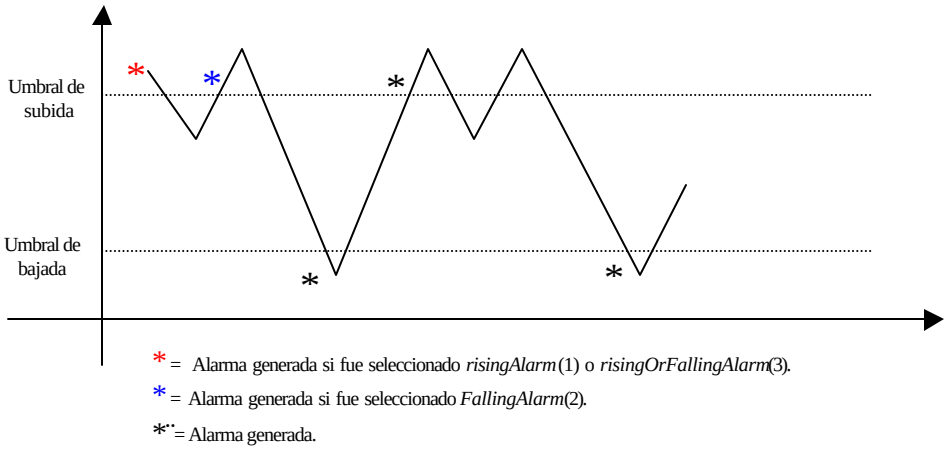

Figura 4.4: Ejemplo de generación de alarmas

4.6.4 Grupo host

El grupo host se encarga de almacenar información sobre cada uno de los hosts detectados por el explorador.
Este grupo consta de tres tablas, una de control (*hostControlTable*) y dos de datos (*hostTable* y *hostTimeTable*). Por cada fila nueva creada en la tabla de control, el monitor creará una nueva fila en las tablas de datos por cada host nuevo encontrado en la subred seleccionada en la tabla de control. El explorador RMON reconoce nuevos hosts en una LAN observando las direcciones MAC de fuente y destino que son enviadas a través de la red.
Las dos tablas de datos están conformadas por los mismos datos (los datos en el explorador RMON pueden estar almacenados una sola vez y accederse a los mismos mediante dos métodos lógicos diferentes) sólo que se presentan al gestor ordenados mediante índices diferentes. En el caso de la tabla *hostTable* se ordenan dependiendo de la dirección MAC del host. En cambio, en la tabla *hostTimeTable* se ordenan por el orden en que el monitor detecta los hosts de la subred.
Una de las ventajas de la tabla *hostTimeTable* radica en que el gestor no necesita recorrer la tabla completa para conocer si se ha encontrado un nuevo host.
Los datos almacenados en la tabla de datos son los siguientes:

- Paquetes entrantes.
- Paquetes salientes.
- Octetos entrantes.
- Octetos salientes.
- Paquetes erróneos enviados por el host.
- Paquetes broadcast enviados por el host.
- Paquetes multicast enviados por el host.

4.6.5 Grupo hostTopN

El grupo hostTopN muestra los primeros N hosts de una lista ordenados en función de algún parámetro. Por ejemplo, se podrían mostrar los 10 hosts que generaron mayor cantidad de paquetes erróneos durante un determinado tiempo.
Este grupo está formado por una tabla de control (*hostTopNControlTable*) y una tabla de datos (*hostTopNTable*).
Para que el explorador RMON recolecte información de este grupo, el gestor debe crear una fila en la tabla de control con los datos siguientes:

- El objeto perteneciente a la tabla *hostTable* del grupo *host* mediante el cual se quiera hacer la clasificación. Para este objeto se define el siguiente tipo de datos:

```
INTEGER{hostTopNInPkts(1),
        hostTopNOutPkts(2),
        hostTopNInOctets(3),
        hostTopNOutOctets(4),
        hostTopNOutErrors(5),
        hostTopNOutBroadcastPkts(6),
        hostTopNOutMulticastPkts(7)}
```

- El número máximo (N) de hosts que se deben presentar en la tabla.
- Duración del intervalo de medición.

Para definir el intervalo de medición debe asignarse un valor al objeto *hostTopNTimeRemaining*. Este objeto se irá decrementando una unidad por segundo mostrando el tiempo restante para que la información en la tabla de datos esté disponible. Cuando se escribe el objeto *hostTopNTimeRemaining*, el valor también es copiado al objeto *hostTopNDuration* (que es de sólo escritura). Este último objeto es el que muestra la duración del intervalo de medición para el reporte.

Una vez que el gestor creó la nueva fila en la tabla de control, el explorador debe leer el parámetro requerido en la tabla del grupo *host* al comienzo y al final del intervalo de medición. Luego compara la diferencia de estos dos valores entre todos los hosts y almacena los primeros N hosts en la tabla de datos.
La tabla de datos posee los datos siguientes: el índice que hace referencia a la tabla de control, el índice propio de la tabla, la dirección MAC del host y el parámetro con el cual se está realizando la clasificación.
Si el gestor requiere realizar un nuevo intervalo de medición debe copiar el valor del objeto de *hostTopNDuration* en el objeto *hostTopNTimeRemaining*. De esta manera el monitor elimina las filas en la tabla de datos correspondientes a los valores de la medición anterior y cuando haya pasado el tiempo del intervalo creará las filas con los nuevos datos recolectados.

4.6.6 Grupo matrix

El grupo Matrix recolecta información sobre el tráfico entre dos hosts dentro de una subred.
La información se guarda con el formato de una matriz, como se muestra en la figura 4.5.
Este grupo esta conformado por una tabla de control (*matrixControlTable*) y dos tablas de datos (*matrixSDTable* y *matrixDSTable*).

		Dirección de destino				
		M(i,1)	**M(i,2)**	**M(i,3)**	**...**	**M(i,N_i)**
Dirección de la fuente	**M(i,1)**	-	R	R		R
	M(i,2)	R	-	R		R
	M(i,3)	R	R	-		R
	...					
	M(i,N_i)	R	R	R		-

R = fila que se encuentra en ambas tablas de datos.
M(i,j) < M(i,j+1)

Figura 4.5: Vista lógica de las dos tablas de datos asociadas a la fila i de la tabla de control

El gestor debe crear una nueva fila en la tabla de control incluyendo la interfase de subred de la cual desea obtener la información. Cuando el monitor reconoce la nueva fila en la tabla de control genera las filas necesarias en ambas tablas de datos. Las dos tablas de datos poseen la misma información organizada de dos formas diferentes:

- Dirección MAC de fuente.
- Dirección MAC de destino.
- Índice de la tabla.
- Paquetes enviados de fuente a destino (*matrixSDTable*) o viceversa (*matrixDSTable*).
- Octetos enviados de fuente a destino (*matrixSDTable*) o viceversa (*matrixDSTable*).
- Paquetes erróneos enviados de fuente a destino (*matrixSDTable*) o viceversa (*matrixDSTable*).

La tabla *matrixSDTable* está ordenada en primer lugar por el índice de tabla, luego por la dirección de fuente y por último con la dirección de destino. En cambio, la tabla *matrixDSTable* está ordenada en primer lugar por el índice de tabla, luego por la dirección de destino y por último con la dirección de fuente.

4.6.7 Grupo filter

Este grupo provee las herramientas para que una estación de gestión pueda instruir a un monitor para que observe algún grupo de paquetes seleccionados o a una interfase en particular (y de esta manera a una subred específica). Se definen dos tipos de filtros:

- Filtro de datos: este tipo de filtro permite al monitor observar a los paquetes en función de un patrón de bits. Puede seleccionarse el filtro para que los paquetes que pasen el filtro sean los que coincidan con el patrón de bits, o los que no coincidan.
- Filtro de estado: este tipo de filtro permite al monitor observar los paquetes con algún estado determinado, como ser: válido o error CRC.

Los filtros pueden combinarse utilizando las operaciones lógicas AND y OR para poder crear filtros más complejos.
Mediante la creación de un conjunto de filtros se genera un *canal* por el cual pueden pasar únicamente los paquetes que pasen la prueba del filtro, manteniéndose una cuenta de la cantidad de estos paquetes. Puede configurarse al monitor de manera tal que cuando algún paquete pase a través del canal genere un evento definido en el grupo *event* o que el paquete sea capturado en caso de encontrarse habilitado el grupo *capture*.

4.6.7.1 Definición y funcionamiento de un canal

Los filtros que conforman un canal pueden combinarse de dos maneras diferentes, pudiendo seleccionarse mediante el objeto *channelAcceptType*. Los valores que puede tomar este objeto son: *acceptMatched*(1) o *acceptFailed*(2).
Si el valor del objeto es *acceptMatched*, los paquetes sólo serán aceptados por el canal si pasan a través de los filtros de datos y de estado de alguno de los filtros asociados al canal. En cambio, si el valor del objeto es *acceptFailed*, los paquetes serán aceptados por el canal si no atraviesan el filtro de datos o el filtro de estado en cada uno de los filtros asociados al canal.
Esta relación lógica puede verse gráficamente en la figura 4.6. Para esto se define como un 1 lógico cuando un paquete puede atravesar el filtro y como un 0 lógico cuando no puede hacerlo.

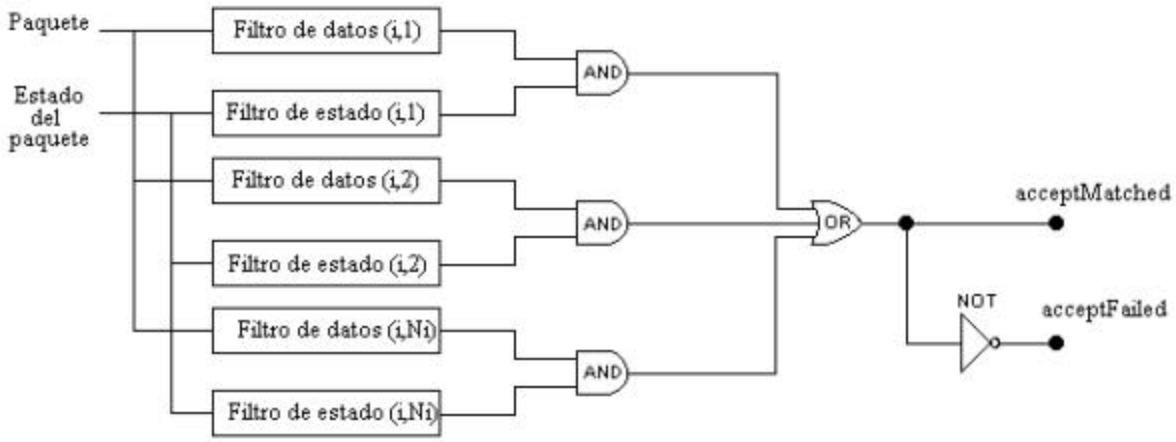

Figura 4.6: Filtro lógico para el canal i.

4.6.7.2 Estructura del grupo filter

El grupo filter está formado por dos tablas de control: *channelTable* y *filterTable*. Cada fila de la tabla *filterTable* define un nuevo filtro y cada fila en la tabla *channelTable* define un nuevo canal asociado con uno o más filtros definidos en *filterTable*.
En la tabla de filtros se pueden definir los siguientes objetos:

- Número del canal al que está asociado el filtro.
- Desplazamiento en bits desde el comienzo del paquete hasta donde debe comenzar el proceso de comparación de bits.
- Bits que deben compararse con cada paquete.
- Máscara que se debe aplicar al proceso de comparación para el filtro de datos.
- Máscara de inversión (inversion mask) que debe aplicarse en el proceso de comparación para el filtro de datos.
- Estado que deben tener los paquetes para pasar el filtro de estado.
- Máscara que se debe aplicar al proceso de comparación para el filtro de estado.
- Máscara de inversión (inversion mask) que debe aplicarse en el proceso de comparación para el filtro de estado.

En la tabla de canal están definidos los siguientes objetos:

- *channelAcceptType*: este objeto define el tipo de canal que puede ser *acceptMatched* o *acceptFailed*.
- *channelDataControl*: si el valor de este objeto se encuentra en *on*, los datos, el estado y los eventos pasan a través del canal. En cambio, si se encuentra en *off*, ningún dato, estado o evento pasará a través del canal.
- *channelTurnOnEventIndex*: cuando sea generado el evento que especifica este objeto se cambiará el estado de *channelDataControl* a *on*.
- *channelTurnOffEventIndex*: cuando sea generado el evento que especifica este objeto se cambiará el estado de *channelDataControl* a *off*.
- *channelEventIndex*: Especifica el evento con el cual está asociado el canal. Este se define mediante el índice que posea el evento en la tabla del grupo *event*.
- *channelEventStatus*: Si hay algún evento asociado al canal la interpretación de este objeto es la siguiente:
  - *eventReady*(1): se genera un evento y luego pasa al estado *eventFired*(2).
  - *eventFired*(2): en este estado no se generan eventos.
  - *eventAlwaysReady*(3):todos los paquetes generan un evento.
- *channelMatches*: cantidad de paquetes que atravesaron el canal.
- *channelDescription*: Descripción textual del canal.

4.6.8 Grupo packet capture

Este grupo se usa para configurar la captura de paquetes de uno de los canales definidos en el grupo *filter* y está formado por una tabla de control *bufferControlTable* y una tabla de datos *captureBufferTable*.
Cada fila de la tabla de control define un buffer que será utilizado para almacenar los paquetes capturados. Pueden modificarse las siguientes características del buffer:

- Máximo número de octetos que se podrán almacenar de cada paquete a partir del inicio del mismo.
- Máximo número de octetos que se devolverán por vez en cada pedido SNMP de un paquete.
- Máximo número de octetos que puede almacenar el buffer. Si se coloca el valor -1, el buffer almacenará la mayor cantidad de octetos posible.
- Indicación del número de paquetes almacenados en el buffer.
- Puede configurarse el buffer de dos maneras diferentes:

- Una vez que esté lleno se eliminará el paquete más antiguo para almacenar el capturado recientemente.
- una vez que esté completo no se capturan más paquetes.

En cada fila de la tabla de datos se almacena el paquete capturado, el tiempo transcurrido en milisegundos desde que se inició la captura de paquetes hasta que fue capturado el paquete de la fila, y la longitud del paquete recibido (puede no estar almacenado todo el paquete, esto depende del tamaño máximo definido en la tabla de control).

4.6.9 Grupo event

El grupo event define los eventos a ejecutarse luego de generada una alarma del grupo alarm.
Este grupo está formado por una tabla de control (*eventTable*) y una tabla de datos (*logTable*). En la tabla de control cada fila representa un evento diferente. Los eventos que pueden generarse son:

- Ninguno.
- Enviar un Trap.
- Archivar el evento en la tabla de datos.
- Enviar un Trap y archivar el evento.

Los datos que se almacenan en la tabla de datos cada vez que se archiva un evento son:

- Índice que hace referencia al evento en la tabla de control que generó esta fila.
- El tiempo en que se generó el evento (muestra la variable *sysUpTime*).
- Descripción del evento generado (depende de la implementación del monitor).

4.6.10 Grupo tokenRing

Este grupo está definido completamente en la RFC 1513 y está dividido en 4 subgrupos:

1. `TokenRing Ring Station`.
2. `TokenRing Ring Station Order`.
3. `TokenRing Ring Station Configuration`.
4. `TokenRing Ring Source Routing`.

4.6.10.1 TokenRing Ring Station.

Este grupo contiene las estadísticas y el estado de todas las estaciones token ring en el anillo local. Además, provee información sobre el estado de cada uno de los anillos que están siendo monitoreados. Este grupo está formado por una tabla de control (*ringStationControlTable*) y por una tabla de datos (*ringStationTable*).
La tabla de control posee una fila por cada subred detectada por el explorador RMON y la tabla de datos posee una fila por cada host detectado dentro de cada una de las subredes descriptas en la tabla de control.

4.6.10.2 TokenRing Ring Station Order

Este grupo posee una única tabla (*ringStationOrderTable*) que representa la ubicación de cada estación dentro del anillo en el que se encuentra.
Para determinar la ubicación se define el objeto *ringStationOrderOrderIndex*. El valor de este objeto es un número entero que representa la cantidad de saltos más uno desde el explorador RMON hasta la estación de trabajo.

4.6.10.3 TokenRing Ring Station Configuration

Este grupo gestiona las estaciones token ring por medios activos. Cualquier estación en un anillo gestionado puede ser removida y se puede obtener información sobre la configuración de la misma. Estas dos opciones pueden ser llevadas a cabo mediante los objetos definidos en este grupo.
Este grupo consta de una tabla de control (*ringStationConfigControlTable*) y una tabla de datos (*ringStationConfigTable*).

4.6.10.4 TokenRing Ring Source Routing

Este grupo contiene estadísticas de la utilización de la información de source routing presente (opcionalmente) en los paquetes token ring.
Este grupo contiene una única tabla (*sourceRoutingStatsTable*) con una entrada por cada interfase.