

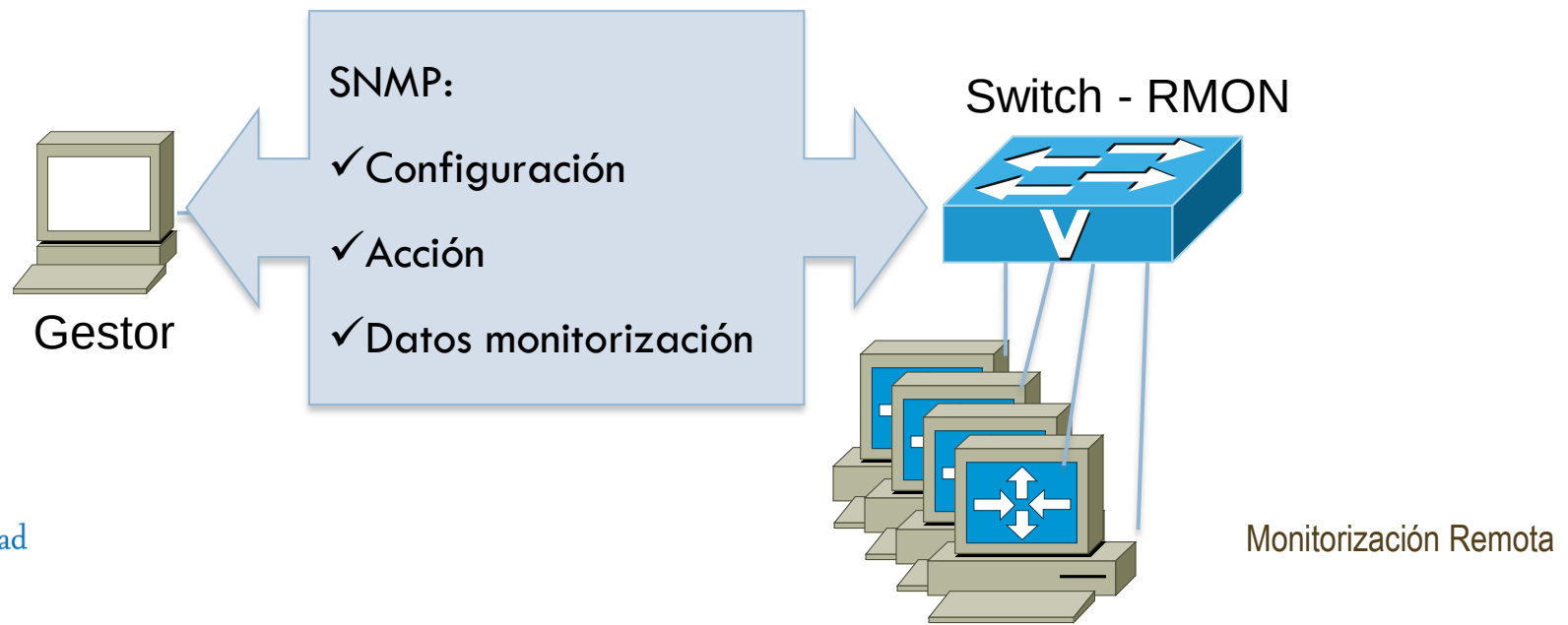


Monitorización Remota

Gestión de Redes

- RMON: ampliación de SNMP, SMI, MIB para monitorización remota.
- Documentos:
 - ▣ RFC2819: RMON-1 de monitorización remota.
 - ▣ RFC4502: RMON-2 de monitorización remota.
 - ▣ Otros (RFC3577): SMON, TPM, APM, RAQMON, DSMON
- MIB que define funciones de monitorización e interfaz.
- Utiliza el protocolo SNMP.

- Monitor que obtiene información de los interfaces de red:
 - ▣ Un dispositivo dedicado (sonda).
 - ▣ Una función disponible en un sistema con recursos (memoria y procesador) dedicados (router, switch...).
- Puede realizar tareas más complejas que un agente.



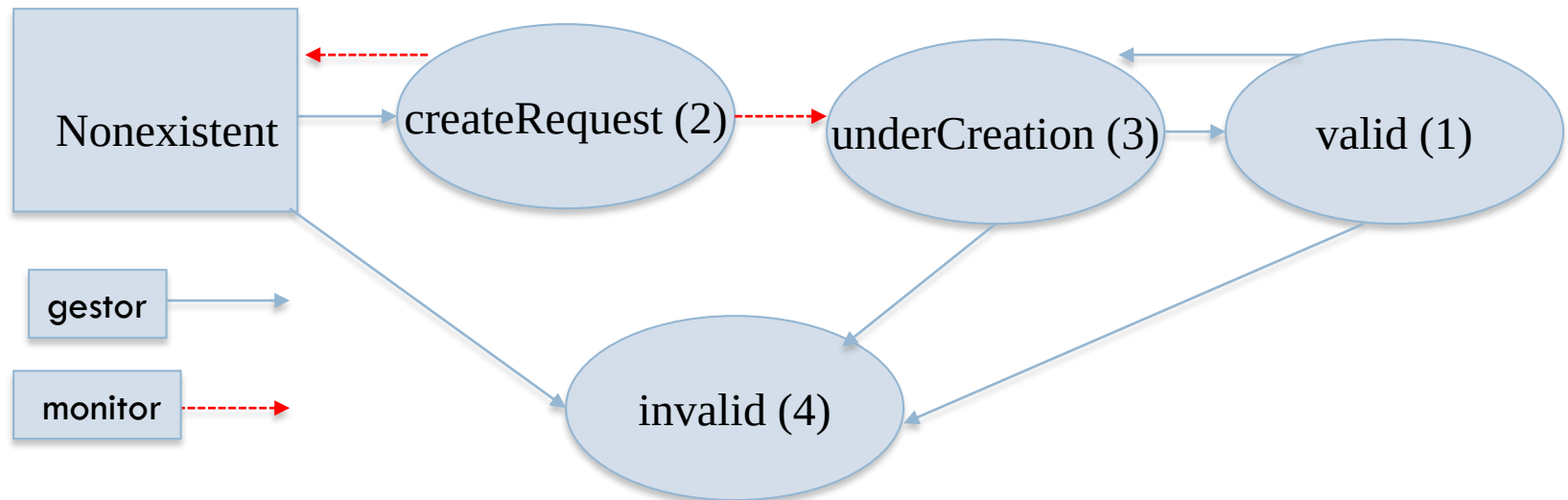
Configuración y obtención de información de monitorización:

- Especifica el tipo de información monitorizada y la forma de hacerlo.
- MIB se organiza en grupos.
- Cada grupo tiene tablas de control (RW) y tablas de datos (RO) asociadas.
- Los parámetros de cada monitorización se especifican en una fila de la tabla de control.
- Los datos recopilados se guardan en filas de las tablas de datos.
- Modificar parámetro en tabla de control: borrar y crear uno nuevo.
- En casos sencillos datos y control pueden estar en la misma tabla.

Invocación de acción:

- Hay objetos de RMON que representan estados
- Se invoca un determinado proceso cuando se cambia de estado (setRequest())

- Se define un procedimiento para modificar, borrar y añadir filas a una tabla.
- En la especificación de RMON se definen **dos nuevos tipos ASN.1**:
 - ▣ Indica el propietario de cada fila de una tabla de RW (control o control/datos):
 - **OwnerString ::= DisplayString**
 - ▣ Indica el estado en que se encuentra cada fila de una tabla RW
 - **EntryStatus ::= INTEGER {valid(1), createRequest(2), underCreation(3), invalid(4)}**
 - ▣ Estos dos parámetros van a existir en todas las **tablas de control**



□ Añadir filas

- El gestor envía una PDU SetRequest con una lista de objetos de una fila y sus valores.
- El monitor recibe la petición y comprueba si los parámetros recibidos no están sujetos a restricciones de MIBs RMON o implementación.
- Si no es posible responde con GetResponse(badValue)
- **Conflicto:** Dos o más gestores intentan crear una fila con los mismos valores. Sólo la primera tiene éxito. Veamos los pasos seguidos:
 - El gestor intenta crear una nueva fila, si no existe otra con el mismo índice, poniéndola con status=createRequest
 - Completada la operación de creación el monitor pone status=underCreation
 - Cuando el gestor ha terminado de crear todas las filas de una configuración las pone con status=valid.
 - Si se intenta crear una fila con status=createRequest y ya existe, el monitor envía un mensaje de error al gestor.

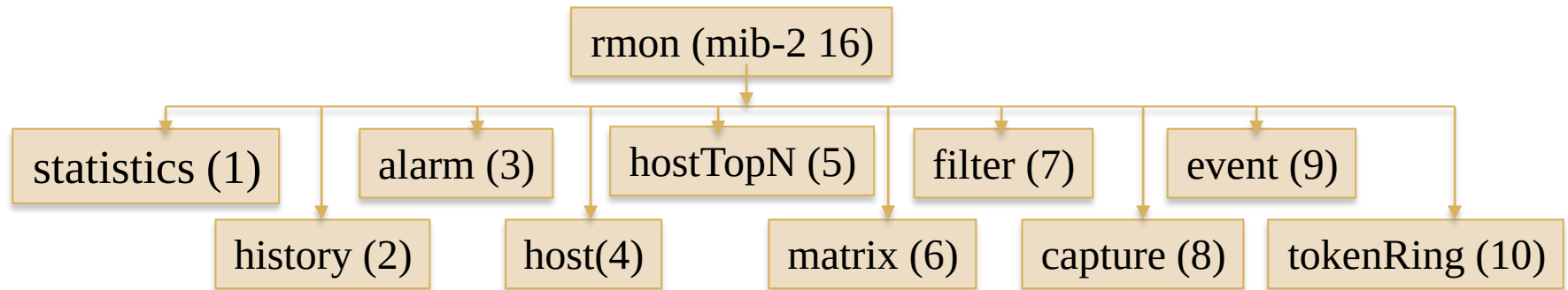
□ **Borrar filas**

- ▣ Se pone EntryStatus=invalid.
- ▣ Solo puede borrar una fila el propietario de esta mediante SetRequest.

□ **Modificar valores**

- ▣ Primero se borra la fila: SetRequest(EntryStatus=invalid).
- ▣ Se crea una nueva fila con los nuevos valores de los parámetros.

MIB RMON



- ❑ **statistics:** Contadores de paquetes con algún tipo de error, octetos y paquetes de distintos tipos.
- ❑ **history:** Toma muestras para ver la evolución de contadores de *statistics* con el tiempo.
- ❑ **alarm:** Monitorización y definición de umbral de alarma de algún OID de las MIBs RMON (requiere *event*).
- ❑ **host:** Contadores de tipos de tráfico por cada host de la subred.
- ❑ **hostTopN:** Los N hosts que tienen un contador con mayor valor (requiere *host*).
- ❑ **matrix:** Información de tráfico entre parejas de hosts.
- ❑ **filter:** Define filtros para capturar o hacer estadísticas de paquetes.
- ❑ **capture:** Captura paquetes definidos por un filtro (requiere *filter*).
- ❑ **event:** Tabla de eventos generados ante condiciones de alarma o contador de filtros que supera un umbral.
- ❑ **tokenRing:** Estadísticas y configuración de subred TokenRing.

- Decodifica paquetes de las capas 3 a 7 del modelo OSI.
 - ▣ Visibilidad a nivel de red. Observa cabeceras nivel de red (normalmente IP) para detectar cosas como:
 - Sobrecarga de LAN por tráfico externo entrante y su origen.
 - Sobrecarga de router por trafico interno saliente y su origen.
 - Sobrecarga por tráfico que atraviesa la red y su origen.
 - ▣ Visibilidad a nivel de aplicación. Observa cabeceras de niveles altos (tcp, http, ftp, e-mail...).

- RMON2 es una extensión de RMON que añade nuevos grupos:
 - **Directorio de protocolos (protocolDir):** Directorio de los protocolos que la sonda puede interpretar.
 - **Distribución de protocolos (protocolDist):** Estadísticas de tráfico generado por protocolo en la LAN.
 - **Mapa de direcciones (addressMap):** Asocia cada dirección de red con MAC y puerto de dispositivo.
 - **Host de nivel de red (nlHost):** Estadísticas de tráfico en hosts identificados por la dirección de red.
 - **Matriz de nivel de red (nlMatrix):** Estadísticas de tráfico entre pares de hosts identificados por dirección de red.
 - **Host de Aplicación (alHost):** Estadísticas de tráfico en hosts identificados por dirección de aplicación (protocolos de capas entre 4 y 7).
 - **Matriz de Aplicación (alMatrix):** Estadísticas de tráfico en pares de hosts identificados por la dirección de aplicación.
 - **Historial de usuario (usrHistory):** Periódicamente muestrea parámetros de variables y logs definidos de usuario.
 - **Configuración (probeConfig):** Define parámetros de configuración estándar para la sonda.

- Indexación de tablas con objetos externos.
 - ▣ En RMON1 las tablas de datos se indexan con:
 - Un índice local que coincide con el de la tabla de control correspondiente.
 - Y después el índice particular de esta tabla.
 - ▣ En RMON2 las tablas de datos se indexan:
 - El índice de la tabla de control correspondiente (externo).
 - Y después el índice particular de esta tabla.

- Indexación de tablas por filtro de tiempo.
 - El gestor hace un sondeo periódicamente a las sondas para obtener valores de las tablas (monitorización).
 - Con un filtro de tiempo se devuelve únicamente los valores que han cambiado desde un instante determinado en la misma petición GetRequest.
 - Para implementar esto se crea un índice en la tabla de datos formado por:
 - **Timestamp:** Marca de tiempo (sysUptime) de la última vez que se actualizó un determinado parámetro (entrada en la tabla).
 - **Index:** Índice que identifica de forma única el parámetro que se monitoriza.

ANEXO: Detalles de RMON v1

Gestión de redes

- Estadísticas sobre una subred Ethernet en forma de contadores de eventos.
- Tabla etherStatsTable con una fila por cada interfaz.
- Algunos eventos de la tabla:
 - Bytes recibidos y paquetes recibidos.
 - Paquetes recibidos dentro de cada uno de los siguientes grupos de tamaño:
 - 64, 65..127, 128..255, 256..511, 512..1023, 1024..1518.
 - Paquetes broadcast y multicast recibidos.
 - Distintos tipos de errores:
 - Pérdidas de paquetes por falta de recursos en la sonda.
 - Error CRC o de alineamiento (nº incorrecto de bytes). Tamaño correcto.
 - Tamaño menor de 64 bytes.
 - Tamaño mayor de 1518 bytes.
 - Tamaño menor de 64 bytes y error de CRC o alineamiento.
 - Tamaño mayor de 1518 y error de CRC o alineamiento.
 - Estimación del nº total de colisiones.
- Objetos no contadores: Índice de la tabla, identificador del interfaz (OID del interfaz en interfaces.ifIndex.valor), propietario y estado de la fila.

- Tabla de Control
- historyControlTable: especifica el interfaz y detalles de la función de muestreo:
 - historyControlIndex: índice de la fila.
 - historyControlSourceData: identificador de la fuente de datos (interfaz).
 - historyControlBucketRequested: número de muestras que se solicita tener guardadas.
 - historyControlBucketGranted: número de muestras guardadas (buffer circular). El monitor intenta que se acerque a BucketRequested lo máximo posible.
 - historyControlInterval: intervalo de muestreo (1..3600) segundos. Por defecto una muestra cada 1800 seg.
 - historyControlOwner y historyControlStatus.

- Guarda los datos de las muestras (muestra por fila):
 - ▣ etherHistoryIndex: índice correspondiente en la tabla de control.
 - ▣ etherHistorySampleIndex: identifica cada muestra de forma única.
 - ▣ etherHistoryIntervalStart: (sysUpTime) comienzo del intervalo de muestreo.
 - ▣ Contadores con la muestra pertenecientes a eventos de etherStatsTable: dropEvents, octets, pkts, broadcastPkts, multicastPkts... Diferencia del valor correspondiente entre el principio y el final del intervalo.
 - ▣ Entero que indica la utilización calculada a partir de etherStatsOctets y etherStatsPkts medidos en un intervalo de tiempo.

- Descubre los hosts en la LAN por direcciones MAC de origen y destino que ve en las tramas (sin error).
- Tres tablas: control (hostsControlTable) y datos (hostTable y hostTimeTable).
- Estructura de hostsControlTable:
 - hostsControlIndex: índice de la tabla. Cada fila identifica un interfaz del monitor. El valor se usa en las otras tablas.
 - hostsControlDataSource: identifica el interfaz por el que se reciben datos de la LAN.
 - hostsControlTableSize: nº de filas en hostTable que referencian esta fila.
 - hostsControlLastDeleteTime: valor de sysUpTime correspondiente a la última vez que una fila de hostTable que referencia a esta fila fue borrada. (0 si no fue borrada ninguna fila).
 - hostsControlOwner y hostsControlStatus.

- Por cada entrada (fila) en la tabla de control hay entradas en la tabla de datos hostTable, una por cada MAC encontrada en el interfaz (LAN).
- Estructura de hostTable: indexada por hostIndex y hostAddress
 - ▣ hostAddress: dirección MAC del host
 - ▣ hostCreationOrder: orden (temporal) de creación del host capturado
 - ▣ hostIndex: valor de hostControlIndex
 - ▣ Contadores:
 - hostIn(Out)Pkts: paquetes transmitidos a/desde el host
 - hostIn(Out)Octets: octetos transmitidos a/desde el host
 - hostOutErrors: paquetes no válidos transmitidos por el host
 - hostOutBroadcastPkts: paquetes difusión transmitidos por el host.
 - hostOutMulticastPkts: paquetes multicast transmitidos por el host.
- Estructura de hostTimeTable: misma información que la anterior pero indexada por hostIndex y hostCreationOrder.

- Mantiene una lista (informe) de los N hosts de una LAN que alcanzan mayor incremento de una variable determinada durante un tiempo de observación.
- Estructura de la tabla de control hostTopNControlTable:
 - HostTopNControlIndex: identifica un informe sobre un interfaz. Se define en una fila de la tabla.
 - HostTopNHostIndex: valor de hostIndex en la tabla de datos del grupo host. Identifica por lo tanto una LAN.
 - HostTopNRateBase: especifica uno de los siete contadores en la tabla de datos del grupo host.
 - HostTopNTimeRemaining: tiempo que queda para finalizar el informe (seg).
 - HostTopNDuration: duración del intervalo de muestreo del informe (seg).
 - HostTopNRequestedSize: máximo número de hosts solicitado en el informe.
 - HostTopNGrantedSize: máximo número de hosts en el informe.
 - HostTopNStartTime: hora (sysUpTime) en que se inicializó el informe la última vez.
 - hostTopNControlOwner y hostTopNControlStatus.

- Estructura de la tabla de datos hostTopNTable:
 - ▣ HostTopNReport: identificador del informe. Mismo valor que HostTopNControlIndex.
 - ▣ HostTopNIndex: índice que identifica una fila de un informe.
 - ▣ HostTopNAddress: Dirección MAC del host.
 - ▣ HostTopNRate: diferencia del valor seleccionado en HostTopNRateBase desde que se inició el informe.

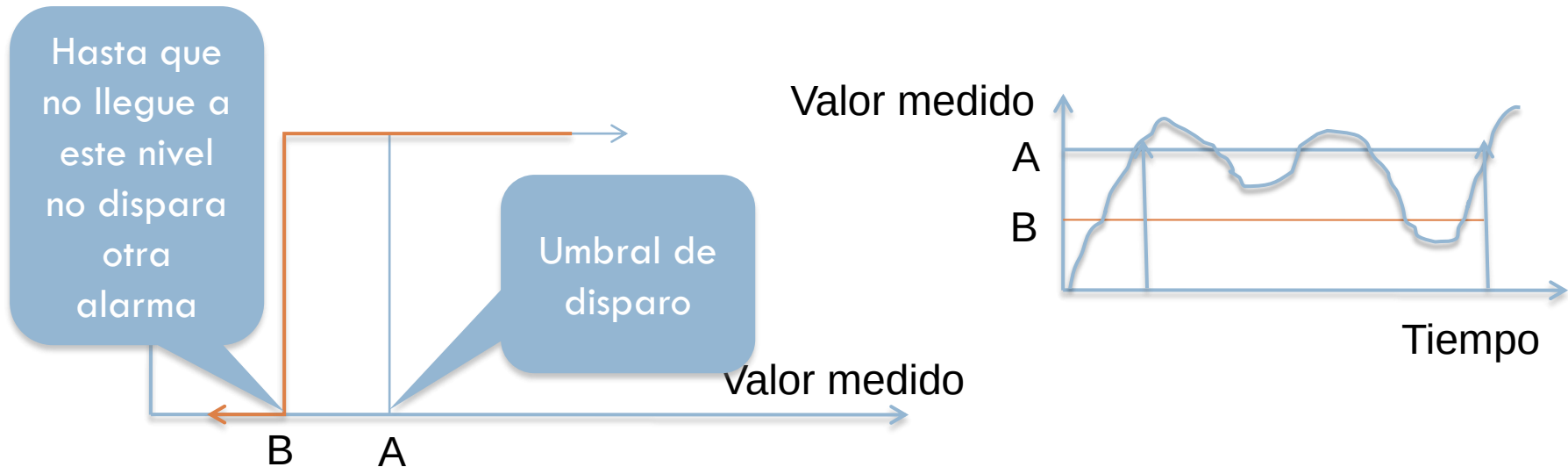
- Proceso de inicio de un informe:
 - ▣ Crear una fila en la tabla de control en la que selecciona un contador.
 - ▣ hostTopNTimeRemaining inicia cuenta descendente hasta que alcanza el valor 0 (fin de intervalo).
 - ▣ Mide la diferencia entre el valor del contador al principio y final del intervalo. Para cada host de la LAN seleccionada.
 - ▣ Lista los hostTopNGrantedSize hosts que tienen mayor el valor medido en orden decreciente.
 - ▣ Para reiniciar el proceso, el gestor debe actualizar hostTopNTimeRemaining con el valor de hostTopNDuration.

- Almacena información sobre tráfico entre cada par de hosts en una LAN.
- Tres tablas: control (matrixControlTable) y datos(matrixSDTable y matrixDSTable).
- Estructura de la tabla de control matrixControlTable:
 - matrixControlIndex: identifica una fila de la tabla que define la comunicación observada en un interfaz.
 - matrixControlDataSource: interfaz o LAN que observa el monitor.
 - matrixControlTableSize: número de entradas en las tablas de datos que se refieren a esta entrada en la tabla de control.
 - matrixControlLastDeleteTime: valor de sysUpTime de la última vez que se borro una entrada en las tablas de datos (0 si no se ha borrado).
 - matrixControlOwner y matrixControlStatus.

- Estructura de la tablas de datos matrixSDTable:
 - ▣ matrixSDSourceAddress: dirección MAC fuente.
 - ▣ matrixSDDestAddress: dirección MAC destino.
 - ▣ matrixSDIndex: identificador. Mismo valor que matrixControllIndex.
 - ▣ matrixSDPkts: paquetes transmitidos de fuente a destino.
 - ▣ matrixSDOctets: octetos transmitidos de fuente a destino.
 - ▣ matrixSDErrors: paquetes transmitidos de fuente a destino, con error.
 - ▣ Indexada con matrixSDIndex, matrixSDSourceAddress y matrixSDDestAddress.
 - ▣ Se ve ordenada la información que va desde una fuente a varios destinos.

- Estructura de la tabla matrixDSTable:
 - ▣ Tiene la misma información.
 - ▣ Indexada con matrixDSIndex, matrixDSDestAddress y matrixDSSourceAddress.
 - ▣ Se ve ordenada la información que va desde varias fuentes a un mismo destino.

- Permite definir umbrales que disparan una alarma si se rebasan.
- Una tabla única (alarmTable).
- Se puede programar alarmas con ciclos de **histéresis** para evitar disparos ante pequeñas fluctuaciones alrededor del umbral.



Estructura de AlarmTable

27

- alarmInterval: intervalo de muestreo en segundos (toma muestra y compara).
- alarmVariable: OID de la variable de la MIB RMON que se monitoriza, puede ser (INTEGER, counter, gauge o TimeTicks).
- alarmSampleType: el método de cálculo del valor que se compara con el umbral, puede ser absoluto (absoluteValue) o diferencial (deltaValue).
- alarmValue: valor de la variable en el último periodo de muestreo.
- alarmStartupAlarm: si se dispara por subida (risingAlarm), bajada (fallingAlarm) o ambos (risingOrFallingAlarm).
- alarmRisingThreshold: umbral para disparo de alarma por subida.
- alarmFallingThreshold: umbral para disparo de alarma por bajada.
- alarmRisingEventIndex/alarmFallingEventIndex: índice del evento llamado (tabla de eventos) cuando se cruza el umbral.
- AlarmOwner y alarmStatus.

- Observa paquetes seleccionados de un interfaz de red.
- Dos tipos de filtros:
 - ▣ Filtro de datos: paquetes que tienen un determinado patrón de bits o que no tienen un determinado patrón.
 - ▣ Filtro de estado: paquetes con un determinado estado (error CRC, longitud menor de 64 bytes o mayor de 1518 bytes...).
- Podemos construir un test con la combinación de varios filtros mediante operaciones lógicas AND y OR.
- Un canal es el flujo de paquetes que pasan un test, hay un contador del número de paquetes del canal.
- Se puede poner un evento si un paquete pasa el canal. (grupo event).
- Se pueden capturar los paquetes de un canal (grupo capture).
- Hay dos tablas de control: filterTable y channelTable.

Capture Group

29

- Captura paquetes de un determinado canal del grupo filtros.
- Tablas de control (bufferControlTable) y de datos (captureBufferTable).
- Estructura de la tabla de control:
 - ▣ bufferControlIndex: identifica una entrada que define la captura de un canal.
 - ▣ bufferControlChannelIndex: índice del canal fuente de los datos channelIndex.
 - ▣ bufferControlFullStatus: dice si el buffer está lleno o le queda espacio.
 - ▣ bufferControlFullAction: comportamiento con buffer lleno (bloqueado/circular).
 - ▣ bufferControlCaptureSliceSize: número de octetos capturados por paquete.
 - ▣ bufferControlDownLoadSliceSize/bufferControlDownLoadOffset: número de octetos/offset devueltos ante una petición snmp.
 - ▣ bufferControlMaxOctectsRequested: máximo tamaño del buffer solicitado.
 - ▣ bufferControlMaxOctectsGranted: máximo tamaño del buffer garantizado.
 - ▣ bufferControlCapturedPackets: nº de paquetes en el buffer actualmente.
 - ▣ bufferControlTurnOnTime: valor de sysUpTime cuando el buffer se inició por primera vez.

- Estructura del buffer de datos:
 - ▣ captureBufferControlIndex: mismo valor que bufferControlIndex.
 - ▣ captureBufferIndex: índice de la tabla, identifica un paquete capturado.
 - ▣ captureBufferPacketID: índice que describe el orden en el que el paquete fue recibido en un interfaz particular.
 - ▣ captureBufferPacketData: datos del paquete.
 - ▣ captureBufferPacketLength: longitud del paquete capturado o porción de éste.
 - ▣ captureBufferPacketTime: tiempo de captura referido al inicio del buffer.
 - ▣ captureBufferPacketStatus: estado de error del paquete.

- Soporta la definición de eventos que pueden ser disparado por una condición definida en la MIB (alarma, captura de un filtro).
- Un evento puede a su vez disparar una acción definida en la MIB:
 - ▣ Guardar información en un log.
 - ▣ Envío de un trap.
- Una tabla de control (eventTable) y una de datos (logTable).
- Estructura de la tabla de control:
 - ▣ eventIndex: indice.
 - ▣ eventDescription: descripción textual.
 - ▣ eventType: puede tomar los valores none, log, snmp-trap o log-and-trap.
 - ▣ Eventcommunity: comunidad para enviar el trap si es de tipo trap.
 - ▣ eventLastTimeSent: valor sysUpTime del último evento de esta entrada generado.
 - ▣ eventOwner y eventStatus.

- Estructura de la tabla de datos (asociada con eventos de tipo log):
 - ▣ logEventIndex: índice del evento asociado (eventIndex).
 - ▣ logIndex: índice de la tabla.
 - ▣ logTime: valor de sysUptime cuando se creo la entrada en la tabla.
 - ▣ logDescription: descripción dependiente de la implementación.