Explicación del concepto trap

Como se ha visto, ?**trap**" es el tipo de mensaje asíncrono que el agente envía al NMS para advertirle acerca de un evento que ha tenido lugar en un dispositivo administrado.

La filosofía de los **trap** es evitar el consumo excesivo de ancho de banda por los mensajes de administración. Por eso se envían sin acuse de recibo, lo cual genera cierta incertidumbre sobre si el mensaje llega con éxito a su destino, Para la comunicación se usa el puerto 162 de UDP. Los mensajes **OnformRequest** quedan más para el uso entre NMS.

El mensaje trap se envía en cualquier momento (asíncrono), en cuanto una variable alcanza un valor determinado.

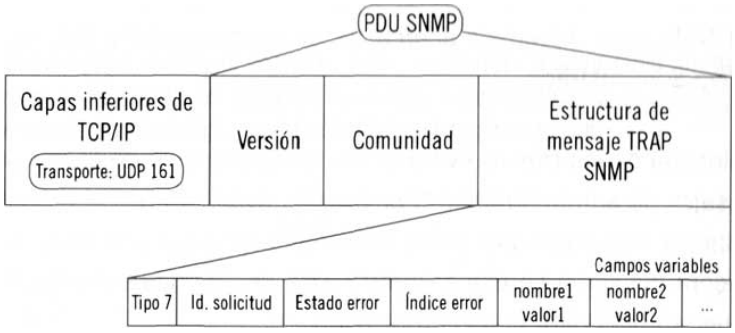
Ejemplo:

El envío de un mensaje de trampa puede producirse cuando una variable alcanza cierto valor por desbordamiento, es el caso de un contador de errores, como **ipInHdrErrors**, o cuando adquiere un valor por cambio de estado, es el caso de una variable de tipo entero, revelando un enlace que se cae, como **ifAdminStatus**. Estos avisos son de gran utilidad para los NMS.

Puede ser además que se tengan multitud de dispositivos administrados y no sea práctico realizar continuas consultas por parte del NMS. Aquí los traps adquieren mayor relevancia.

Las tramas de tipo trap difieren de las normales, tienen el código de tipo 7 y algunos campos diferentes.

Esquema de una trama de tipo trap gnérica



Los campos que incorpora el mensaje:

Empresa: indica el tipo de objeto administrado que provoca la notificación.
Agente: la dirección IP del agente que envía el mensaje.
Código genérico: una primera clasificación de los tipos de traps. Por ejemplo: **Cold start (código 0)** indica un reinicio del agente habiendo podido cambiar la configuración, **Warm start (código 1)** podría indicar un reinicio sin cambio en la configuración, etc.
Código específico: clasificación más concreta sobre el tipo de trap.
Tiempo: el transcurrido desde la última inicialización del agente.
Campos variables: información concreta sobre los valores afectados.

Resumen de los tipos de mensaje de SNMP

Mensaje

Sentido

Operación

GetRequest
NMS --> Agente
Lectura

GetNextRequest
NMS --> Agente
Lectura

GetResponse
Agente --> NMS
Respuesta

SetRequest
NMS --> Agente
Escritura

GetBulkRequest
NMS --> Agente
Lectura

InformRequest
NMS --> NMS y
Notificación

Agente --> NMS

Trap
Agente --> NMS
Notificación

Comparación de las versiones

Existen tres versiones de SNMP que son las siguientes:

SNMPv1. La primera versión incorporaba las funciones GetRequest, GetNextRequest, GetResponse, SetRequest y Trap.

Entre los problemas de esta primera versión figuraba la Imposibilidad de conseguir los datos de una tabla con una sola petición. Se tenían que hacer sucesivos GetRequest y GetNextRequest.

Otro problema era la seguridad. La única manera de establecer cierta se-guridad era a través del nombre de comunidad y los permisos de acceso a los objetos. La comunidad define el dominio de administración del sis-tema SNMP, se trasmite como una contraseña y no es difícil de averiguar pues viaja en texto plano (sin cifrar). Se ve en la obligación de acudir a otras funciones adicionales a SNMP para garantizar la seguridad.

SNMPv2. La versión 2 sigue confiando la seguridad en las comunidades.

En esta versión se modifica el uso de algunos campos de la trama ge-nérica para adaptarla a nuevas funcionalidades. Esas nuevas funciones son GetBulkRequest y InformRequest, y mejorando además, el compor-tamiento de las órdenes de la versión 1.

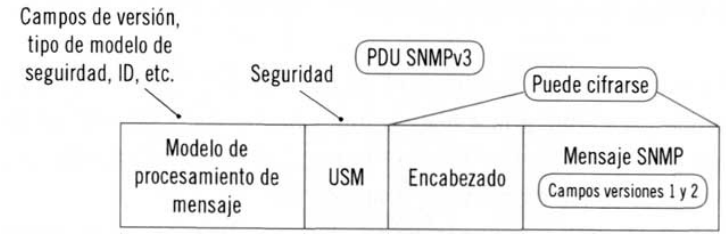
Con GetBulkRequest se consigue hacer peticiones para volúmenes de datos mayores, y así permite que se puedan solicitar tablas sin tener que hacer continuas solicitudes.

Los InformRequest permiten la comunicación entre NMS y que los agen-tes envíen avisos con acuse de recibo.

SNMPv3. Resuelve problemas de seguridad de las versiones anteriores Es como la versión 2, pero con características de seguridad.

Para ello se basa en un modelo de seguridad de usuario con gestión de claves y autenticación (User-based Security Model o USM). Además, el control de acceso a los objetos se hace por un modelo de vistas donde se define a que MIB se puede acceder y con qué permisos. Los mensaje, van acompañados de huellas digitales generadas con una función hash (MD5 o SHA) para garantizar la integridad. También incorpora mecanismos para cifrar la información.
Importante! MD5 (Message Digest) y SHA (Secare Hash Algoritm) son funciones criptográficas especial-mente diseñadas para salvaguardar la integridad en las comunicaciones.
El formato de la trama SNMP se amplía para aportar la seguridad mencionada.

Esquema de una PDU de tipo SNMP3. Se excluyeron los campos de las capas inferiores



Esta versión de SNMP no está siendo masivamente implantada. Se usa más la versión 2. La seguridad que aporta puede ser sustituida por otros métodos, y muchos dispositivos suelen venir con capacidades de SNMPv2.