

Network management applications are outside the scope of the Hub Management standard. However the purpose of providing a Hub Management standard is to provide the necessary hooks to allow network and system management applications to run.

Providing a 32 bit octet counter imposes a restriction on all network and system management applications that they must poll all octet counters within the entire network at least once per hour in order to guarantee the integrity of the values. There could easily be thousands of octet counters in a network.

Consider an application which is running to give advanced warning of network topology changes that are required (eg planting more bridges), say as a result of increasing traffic levels. This would in most networks be adequately served by a process which polls all octet counters once per week and correlates the results over a 3 month sample. This can be done incredibly cheaply (wrt processor power), but becomes expensive if the application is required to poll every counter every hour instead of every week.

We need to find a way of increasing the size of the counters so as not to impose expensive restrictions on network and system management applications, yet at the same time without excluding certain management protocols.

This is the proposed way of doing it:

1. All counters start from zero on power up.
2. All counters are specified in the standard with a size that is large enough to outlive the device. This does not mean they have to be implemented in hardware as this size. The counter specified as a MO attribute is a management view of a resource. The management view has nothing to do with the size of the resource.
3. A comment is added to the standard which enables restrictive management protocols to take a least significant view of the resource. ie 802.1 and ISO can expand the ASN.1 integer up to a maximum of 64 bits, whereas SNMP will reach a ceiling at 32 bits.

Note that taking (1), (2) and (3) together then with 802.1 and ISO the maximum specified size is never reached.

Network management applications are outside the scope of the Hub Management standard. However the purpose of providing a Hub Management standard is to provide the necessary hooks to allow network and system management applications to run.

Providing a 32 bit octet counter imposes a restriction on all network and system management applications that they must poll all octet counters within the entire network at least once per hour in order to guarantee the integrity of the values. There could easily be thousands of octet counters in a network.

Consider an application which is running to give advanced warning of network topology changes that are required (eg planting more bridges), say as a result of increasing traffic levels. This would in most networks be adequately served by a process which polls all octet counters once per week and correlates the results over a 3 month sample. This can be done incredibly cheaply (wrt processor power), but becomes expensive if the application is required to poll every counter every hour instead of every week.

We need to find a way of increasing the size of the counters so as not to impose expensive restrictions on network and system management applications, yet at the same time without excluding certain management protocols.

This is the proposed way of doing it:

1. All counters start from zero on power up.
2. All counters are specified in the standard with a size that is large enough to outlive the device. This does not mean they have to be implemented in hardware as this size. The counter specified as a MO attribute is a management view of a resource. The management view has nothing to do with the size of the resource.
3. A comment is added to the standard which enables restrictive management protocols to take a least significant view of the resource. ie 802.1 and ISO can expand the ASN.1 integer up to a maximum of 64 bits, whereas SNMP will reach a ceiling at 32 bits.

Note that taking (1), (2) and (3) together then with 802.1 and ISO the maximum specified size is never reached.