A lock feature to SNMP
draft-fan-meng-snmp-lock-00

Washam Fan Tony Meng
Huawei Symantec

Background

- SSH has been adopted to transport SNMP.
- SNMP messages including SET might be larger in future.
- A larger SET might need more time for processing, say, more than microseconds. In that case, the larger SET might be intervened by other write operations regardless of the NM interface used.

Problem

- A large SET handling
  - Conflict with write operations regardless of the NM interface used.
- Multiple SETs handling as a transaction
  - Interleave with write operations regardless of NM interface used.
- In above 2 cases, underlying configuration resources need locking to ensure consistency.

Extension to LOCK-MIB

- Managers can create and destroy SNMP locks via manipulating lockSnmpTable. lockSnmpSpinLock is used to coordinate simultaneous SETs to lockSnmpTable

lockSnmpTable

Unsigned32	lockSnmpLockId
SnmpAdminString	lockSnmpViewTreeFamilyViewName
OBJECT IDENTIFIER	lockSnmpViewTreeFamilySubtree
OCTET STRING	lockSnmpViewTreeFamilyMask
INTEGER	lockSnmpViewTreeFamilyType
Unsigned32	lockSnmpIndex
RowStatus	lockSnmpRowStatus

4 lockSnmpViewTreeFamily* fields are jointly used to identify the scope of locked area. Their semantics is specified inRFC3415

Create locks

(1) GET(lockSnmpSpinLock.0) and save in sValue

(2) Determine the scope of locked area

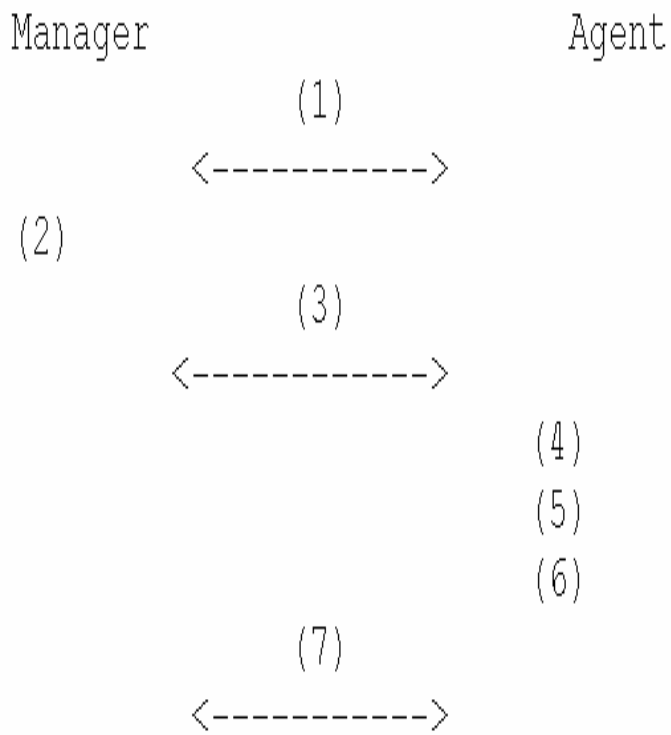
(3) SET(lockSnmpSpinLock.0=sValue, lockSnmpViewTreeFamilyViewName=viewValue, lockSnmpViewTreeFamilySubtree=subtreeValue, lockSnmpVIewTreeFamilyMask=maskValue, lockSnmpViewTreeFamilyViewType=typeValue)

(4) An entry is created with lockSnmpStatus="notReady" by the agent.

(5) Lock request handling

(6) An entry representing the lock is added to lockTable with lockState="ACTIVE" or lockState="FAILED“ depending the result of (5).

(7) GET(lockSnmpRowStatus) for checking if the lock succeeded or failed

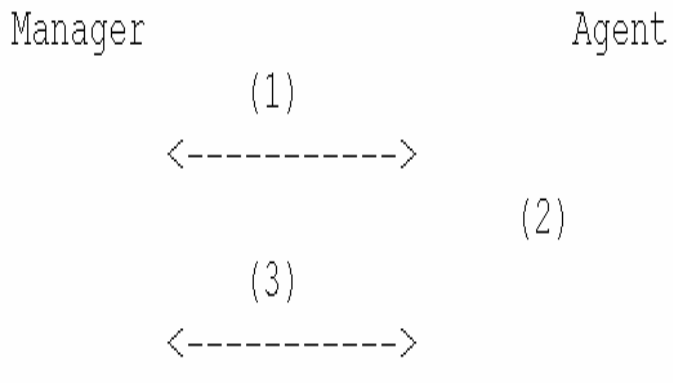


Destroy locks

(1) SET(lockSnmpStatus=destroy)

(2) Unlock request handling (change the corresponding entry in lockTable)

(3) GET(lockSnmpRowStatus) for checking if the unlock succeeded or failed



Impact on SET validation

- According to RFC3416, a SET operation is conceptually processed as a two phase operation. Before actual variable assignments, there are 12 steps for checking.
- If any variable binding’s name specifies an already locked managed objects (or instances), step(11) should be triggered
- I.e., the value of the Response-PDU's error-status field is set to "resourceUnavailable", and the value of its error-index field is set to the index of the failed variable binding.

Security considerations

- A user should have adequate privileges to create SNMP locks.
- Locks held by a user for a long time might prevent other users (regardless of the NM interface used) from configuring the system (which might lead to DoS attack).
- SNMP is not allowed to release non-SNMP locks because of different Access Control Models.
- lock release forcibly might lead to difficult recovery, as It is impossible to rollback all successful SET(s) protected by the lock.

Questions?

- Is it important?
  - A large SET and multi-message SET would be used in real world.
- Is there interest in it?
  - Anyone want to join us on this work?
- Should it be a WG item?

Thanks!

washam.fan@huaweisymantec.com