A Proposal for Access Controls
(aka SAN Boxes)
T10/99-278 revision 2
(Apropos T10/99-245r4)

Jim Hafner
IBM
January 12, 2000
Major Changes from 99-245r2

- Granularity at LU only (no "element" level ACs)
  - 99-245r3 has revised version with "elements"

- Changes in ACs are "indivisible events" wrt other commands

- Added new RESET AC service action (to OUT cmd)
  - Requires check of the Manage ACL Key in normal mode
  - VS-mode for vendor-specific alternatives (e.g., to override the Manage ACL Key)

- Added changes to "Table 8" of SPC-2 for interactions with Reservations

- Small change to "Table 4" of FCP-2 for interactions with fabric events (e.g., LOGO)
Major Changes from 99-245r2 (continued)

- Includes "approved?" new ASC/ASCQs of 99-314r1
  - uses ASC for INVALID COMMAND OPERATION CODE
- TransportID for FCP has 8 bytes for Process Associator
  - 4 for PA and 4 reserved (or is it 7 bytes and 1 reserved? - see fcph3_93, Association Header, fig 52, pg 32)
- Addition of TransportID for SPI (4 bytes, two reserved, two for SCSI Address)
- All other issues from the Nov. teleconf and e-mail addressed
Stuff that stayed from rev2

- Configuration of (non-proxy) ACs requires "Manage ACL Key" shared between configuring application client and LU
- Proxy ACs still available (simple model)
- Persistence
  - via PTPL (optional)
  - non-volatile "constrained" bit required
- Two initiator "ACCESS DENIED" ASCQs
  - INITIATOR NOT ENROLLED (hasn't sent ACCESS ID ENROLL service action)
  - INITIATOR NOT AUTHORIZED (has sent ACCESS ID ENROLL service action but still has no access)
- Plus
  - INVALID MANAGE ACL KEY
Stuff that stayed from rev2 (continued)

- Access granted with
  - AccessID identifier (as enrolled by initiator)
  - TransportID identifier (persistent, e.g., FC-WWN or volatile, e.g., FC-PortID)
Proposed Command Set Summary

- **IN** service actions (Opcode 85h)
  - REPORT ACL (mandatory)
  - REPORT INITIATOR ACL (optional)

- **OUT** service actions (Opcode 86h)
  - ACCESS ID REGISTER (mandatory)
  - MANAGE ACL (mandatory)
  - PROXY ACCESS (optional)
  - RESET AC (optional)

- VS service actions available as well
# ASC/ASCQ Summary

<table>
<thead>
<tr>
<th>ASC</th>
<th>ASCQ</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>20h</td>
<td>01h</td>
<td>ACCESS DENIED - INITIATOR NOT ENROLLED</td>
<td>Initiator has not sent an ACCESS ID ENROLL service action</td>
</tr>
<tr>
<td>20h</td>
<td>02h</td>
<td>ACCESS DENIED - INITIATOR NOT AUTHORIZED</td>
<td>An enrolled initiator has access permissions insufficient for the requested command</td>
</tr>
<tr>
<td>20h</td>
<td>03h</td>
<td>ACCESS DENIED - INVALID MANAGE ACL KEY</td>
<td>The <code>MANAGE ACL KEY</code> value is not valid</td>
</tr>
<tr>
<td>55h</td>
<td>05h</td>
<td>INSUFFICIENT ACCESS CONTROL RESOURCES</td>
<td>The device server has exhausted its resources for access controls</td>
</tr>
</tbody>
</table>
Proxy Model

- All initiators with non-proxy rights are equivalent
  - each can grant or revoke proxy right to third party
- LU does not track who granted right to X
  - A and B granting proxy rights to X is equivalent to A doing it twice
  - minimizes target resource requirements
- Proxy rights are volatile
- Existing Proxies can be clobbered by MANAGE ACL service action
Proxy Model (continued)

- Problems?
  - Unanticipated revocation
    - A and B independently grant proxy to X for some third-party copy operation on AB-shared resource
    - X completes A's job then A revokes proxy for X
    - X can't complete B's job
  - Lingering proxies
    - A grants proxy to X
    - A has its rights revoked
    - X still has proxy right
Proxy Model (continued)

- One possible "solution" is for LU to track the "grantor"
  - is this possible with multiple naming conventions?
  - certainly requires significantly more resources
- Can we solve all "problems" by
  - removing proxy from Access Controls
  - adding function to Extended Copy
- Should we just remove Proxy altogether?
Contacts

- **e-mail**: hafner@almaden.ibm.com
- **phone**: 408-927-1892
- **fax**: 408-927-4182
This is NOT Reservations

- **Reservations**: for cooperating initiators (as we might see in a tightly controlled homogeneous interconnect)
- **Access Controls**: for non-cooperating initiators (as we might see in large heterogenous interconnect)
- Access Controls enable "shared access groups" as subsets of full fabric

[Above simple distinction courtesy of Ralph Weber]
List of Pages from 245r4

- ACCESS CONTROL IN cmd and service actions, Pg 14
- REPORT ACL parameter header, Pg 15, and Entry page, Pg 16
- REPORT INITIATOR ACCESS parameter format, Pg 17
- ACCESS CONTROL OUT cmd and service actions, Pg 18
- MANAGE ACL service action parameter header, Pg 20
- MANAGE ACL and PROXY ACCESS parameter Entry page, Pg 21
- RESET AC service action parameter format, Pg 23
- TransportID for FCP-x, Pg 25; for SPI-x, Pg 27