

Explicit RESET\_CONNECTION  
is *NOT* necessary

**August 16, 1999**  
**Akihiro Shimura**  
**CANON INC.**

## Recovery from aborted task set

- **Current Draft** (PPDT\_r05 as of August 10, 1999) **specifies initiator's two options to recover with aborted target.**
  - Resetting a connection via **RESET\_CONNECTION** control
  - Resynchronizing a connection by utilizing signature and history log
- **RESET\_CONNECTION** control request is issued only from the *initiator*, and the target simply discards the history log upon the request from the initiator.

# What happens?

- **If the queue was idle when aborted,**
  - Safe to issue new ORB for the initiator
  - Safe to issue RESET\_CONNECTION control, but effect is the same as new signature ORB
- **Otherwise, if the queue was active,**
  - I2T queue: there is no way for the target transport to *cancel* the effect on the client
    - I2T case: caused by the data already passed to the client.
    - T2I case: caused by accepting data delivery to the initiator. (the client's data is already stored in the buffer (shared memory on the remote node)).

As a result,

- **Thus, without consulting with the client (that should be *specially* ready for reset) or the initiator, target cannot restore its state regarding data delivery to the state known by the initiator.**
- **As a result,**
  - **I2T case: data to the client may be duplicated.**
  - **T2I case: data from the client may be lost.**

# Examples

