

# *KILL QUEUE* *COMMAND PROPOSAL*

TAKASHI ISODA  
Canon Inc.  
PWG P1394 In San Diego

Dec 14th 1998

1

**Canon**

# The background & problem

*At last meeting there was suggestion with the necessity of kill (queue) command to disconnect the specific queue regardless of its status...*

One of the solutions is Using Abort\_Task\_Set

But it affects all the active queues -> Inefficiency

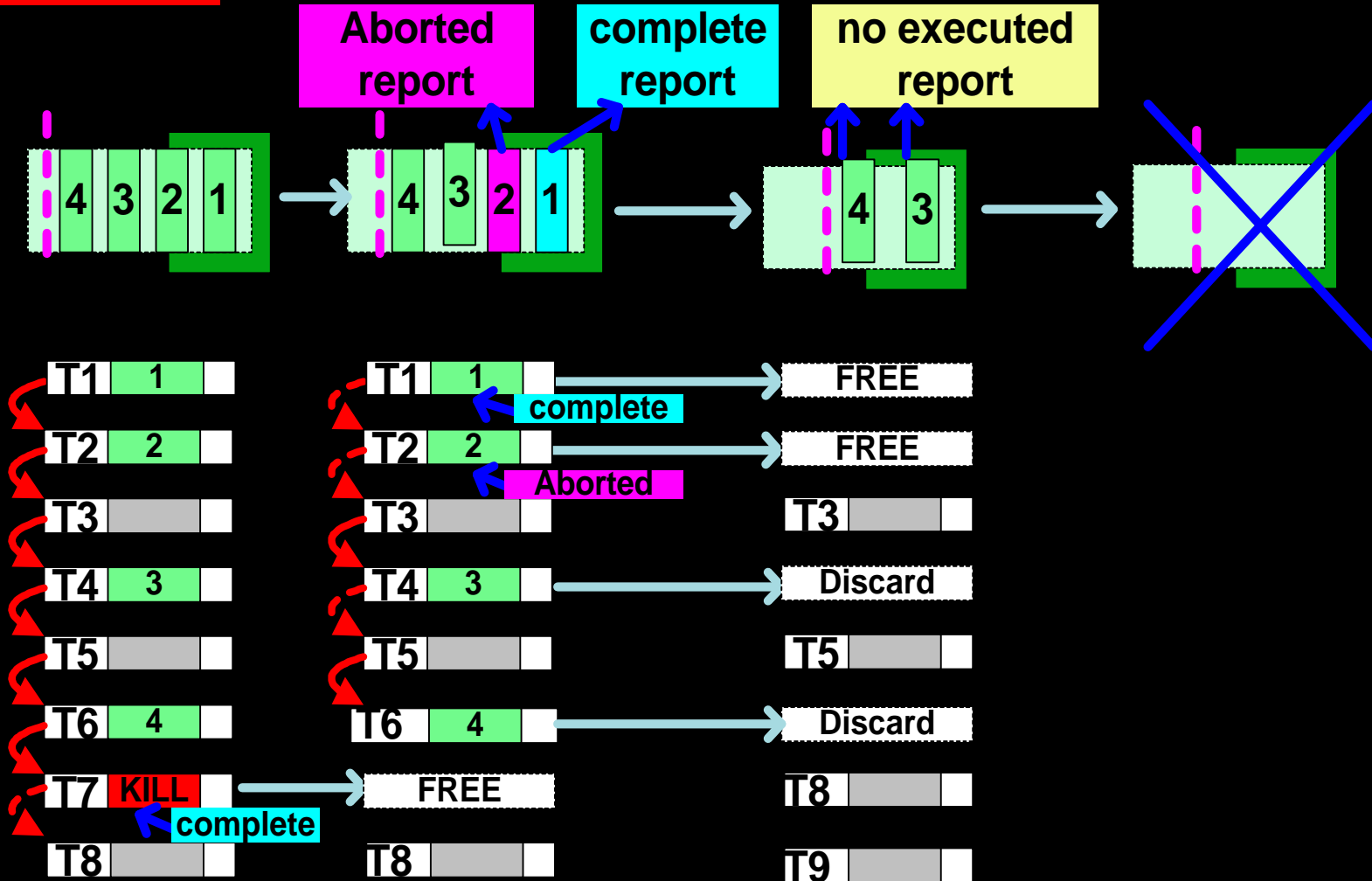
*Current profile is missing the specification of ..*

- **the kill command**
- **the behavior of Target and Initiator for the kill command .**

**We shall define the kill command and the behavior of Target and Initiator for it**

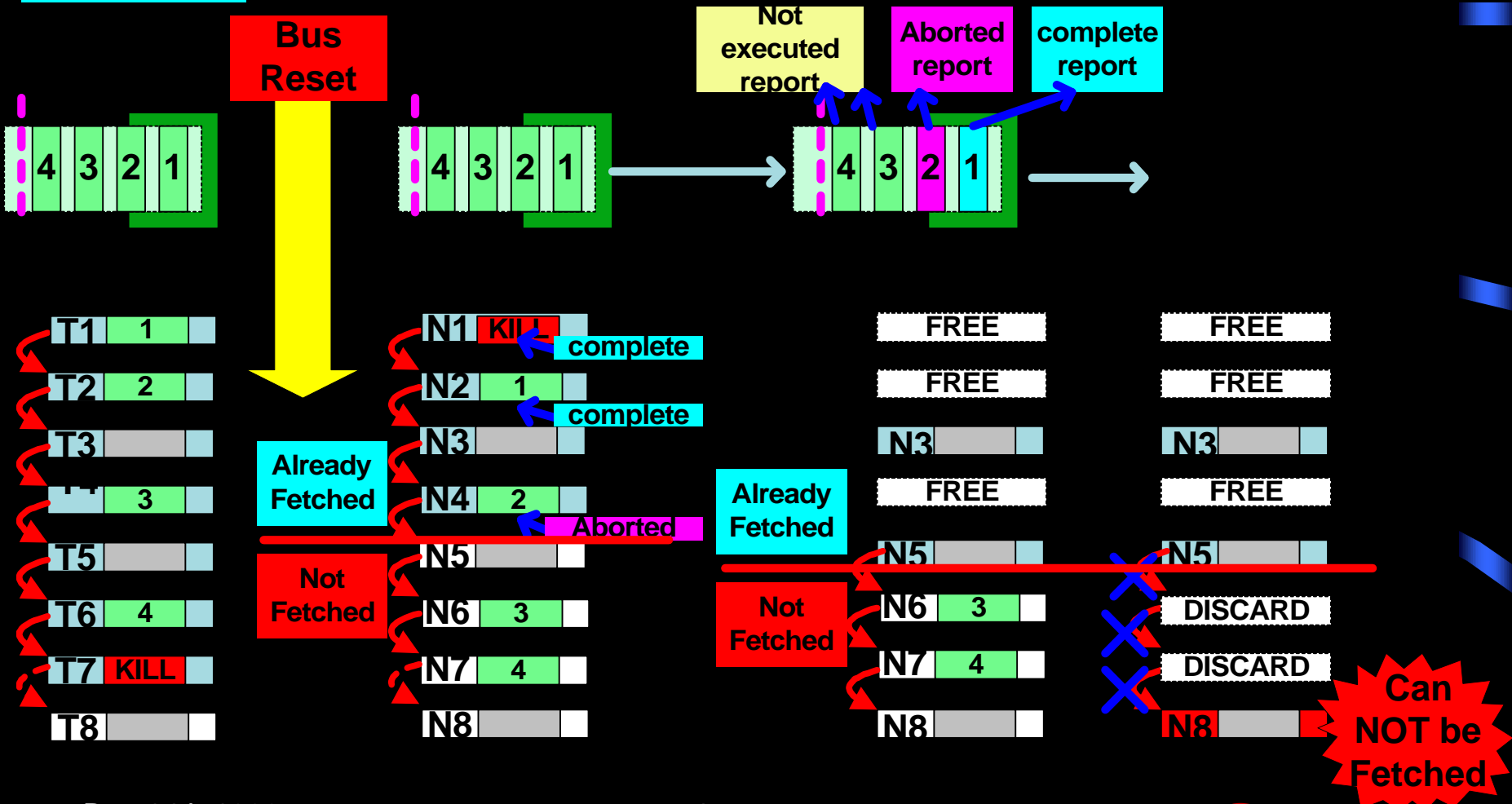
**Kill queue  
command**

*Initiator Shall NOT issue new ORB for the queue*



Without status block for the ORBs after the aborted ORB like Recovery mechatism from Abort Task Set, It would NOT work because....

**Kill queue command** *Initiator Shall NOT issue New ORB for the queue*



# **Solution Candidates(1).....**

## **Solution(1) method**

Clean up the rest of ORBs bounding to the queue that has already killed using completion status without execution.

## **Advantage point**

***NO more mechanism to the implementation is required.***

## **Disadvantage point**

**The complete indication (s) might be redundant essentially**

## **Solution Candidates(2)....**

Make initiator keep the order of ORBs belong to the queue that is been killing and ORBs related to the queue in the queue#0 across Busreset

### **Advantage point**

**No redundant traffic occurs.**

### **Disadvantage point.**

**requires additional mechanism that may conflict with current definition of “queue” to the Initiator implementation.**

**# Queue is an ordered set of ORBs that does not block with respect to other queues**

## ***Which solution is better....***

Disadvantage point of Solution1 is redundant traffic might be occur

***-> In light of whole amount of traffic,  
the volume of redundant traffic can be ignored***

Disadvantage point of Solution2 is to requires additional mechanism that may conflict with current definition of “queue”

***->Not GOOD***

***Solution 1 is better.....***

## *Kill queue command is...*

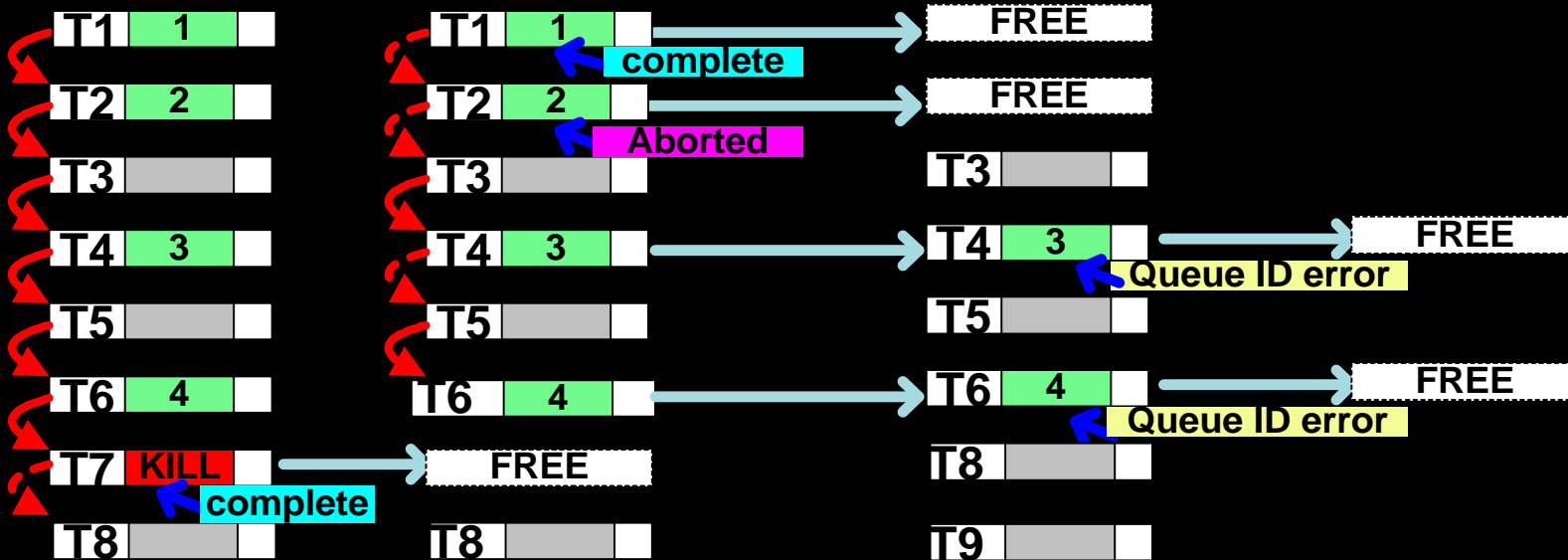
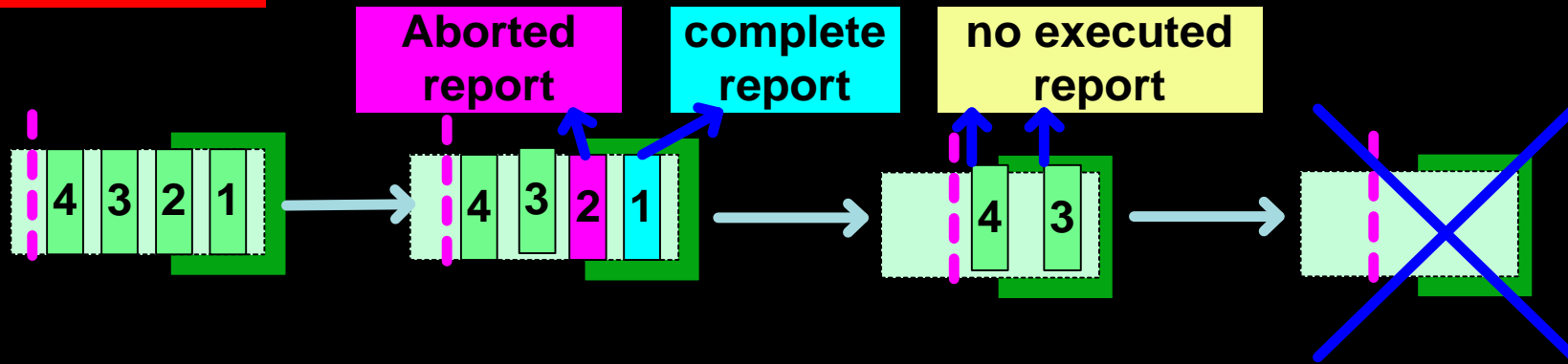
- Abort all the Tasks belong to the specified queue and “disconnect”s the queue
- non graceful disconnect
- Passed through queue#0 (management queue).
- Parameter :
  - queueID(s) of queue(s) that is been killing.
- Not blocked by the flow status of the queue that is been killing.
- Not affects task executions of other queue(s)



# Initiator behavior

**Kill queue command**

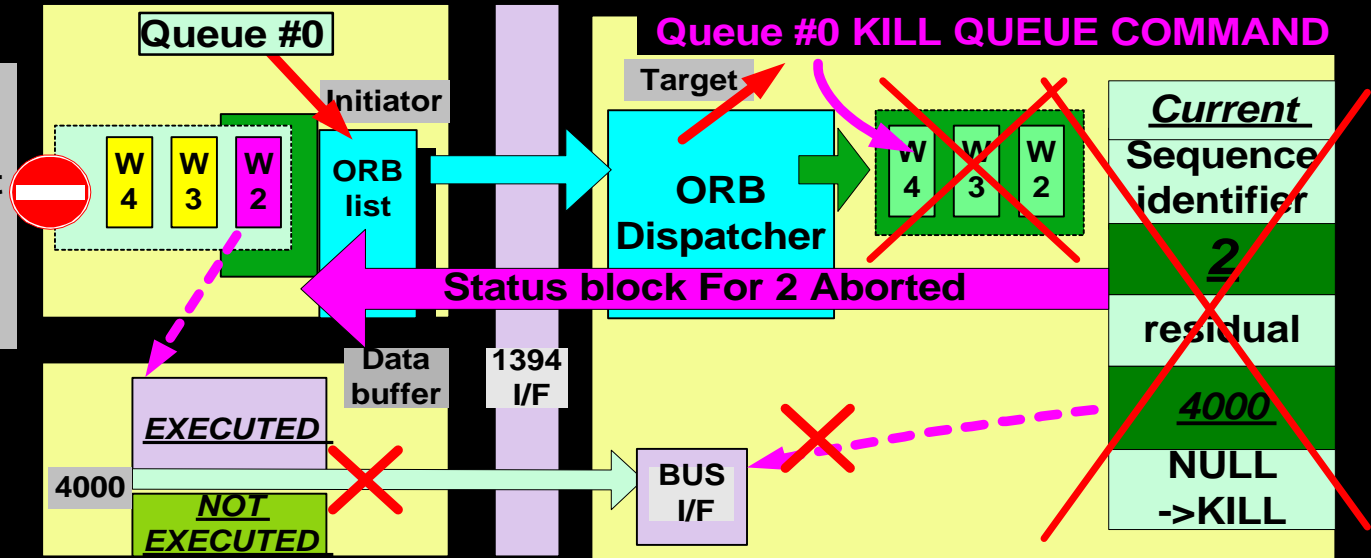
**Initiator Shall NOT issue new ORB for the queue**



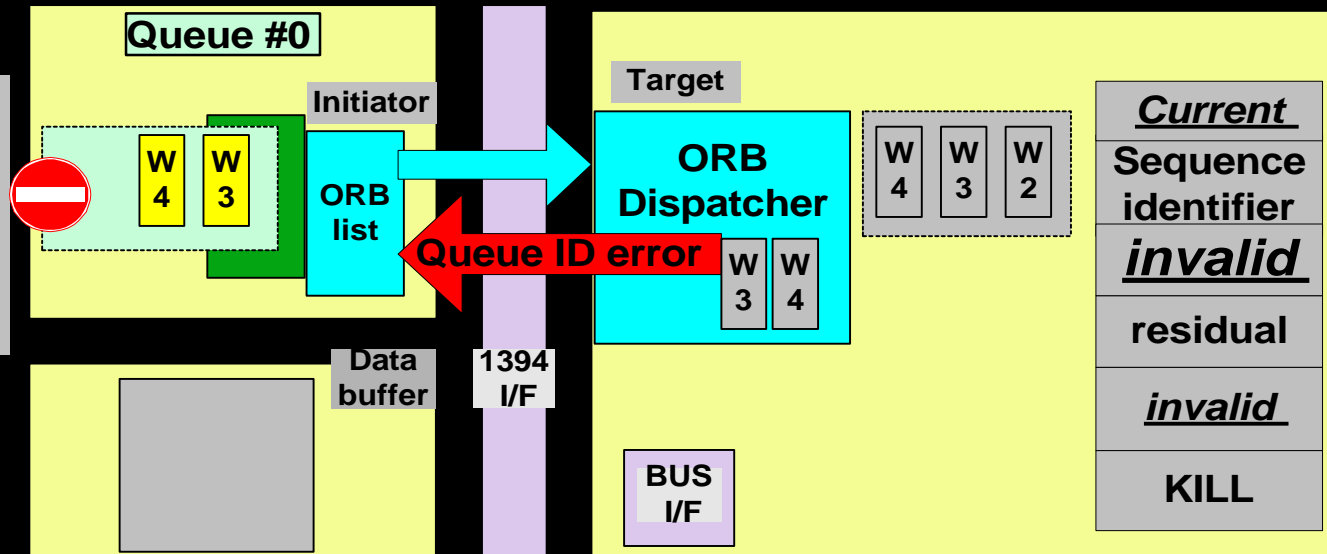
Dec 14th 1998

# Target behavior

Receiving KILL command, the execution agent aborts current execution and kills itself



ORB Dispatcher stores status block with Queue ID error for the ORB bounded for the killed queue



# *Conclusion*

*Solution(1) seems to work well !*  
*How about it?*