Proposal of “Disconnection” Using “Final bit” Flag

Akihiro Shimura

Takashi Isoda

Direction From Maui Meeting..

- Mixing data and control other than disconnect requires new mechanism (synchronize or retry)

  Do we really need control exchange other than disconnect on the data queue?->***No***
- Requirement for disconnect is...
  - To synchronize both end
  - To give a timing to release resources
- To satisfy the requirement…
  - Single flag enough to satisfy the requirement
  - Action based control exchange only increases the redundancy and adds no value

“Disconnect” Proposal Using “Final Bit” Flag

- “Final bit “ is not an action but a flag embedded in both ORBs and status blocks for a data queue.
- The meanings of “final bit “ flag are..
  - In ORB:

    Initiator grants target to release execution resource for this queue
  - In Status Block:

    Target requests initiator to start disconnect process for this queue.

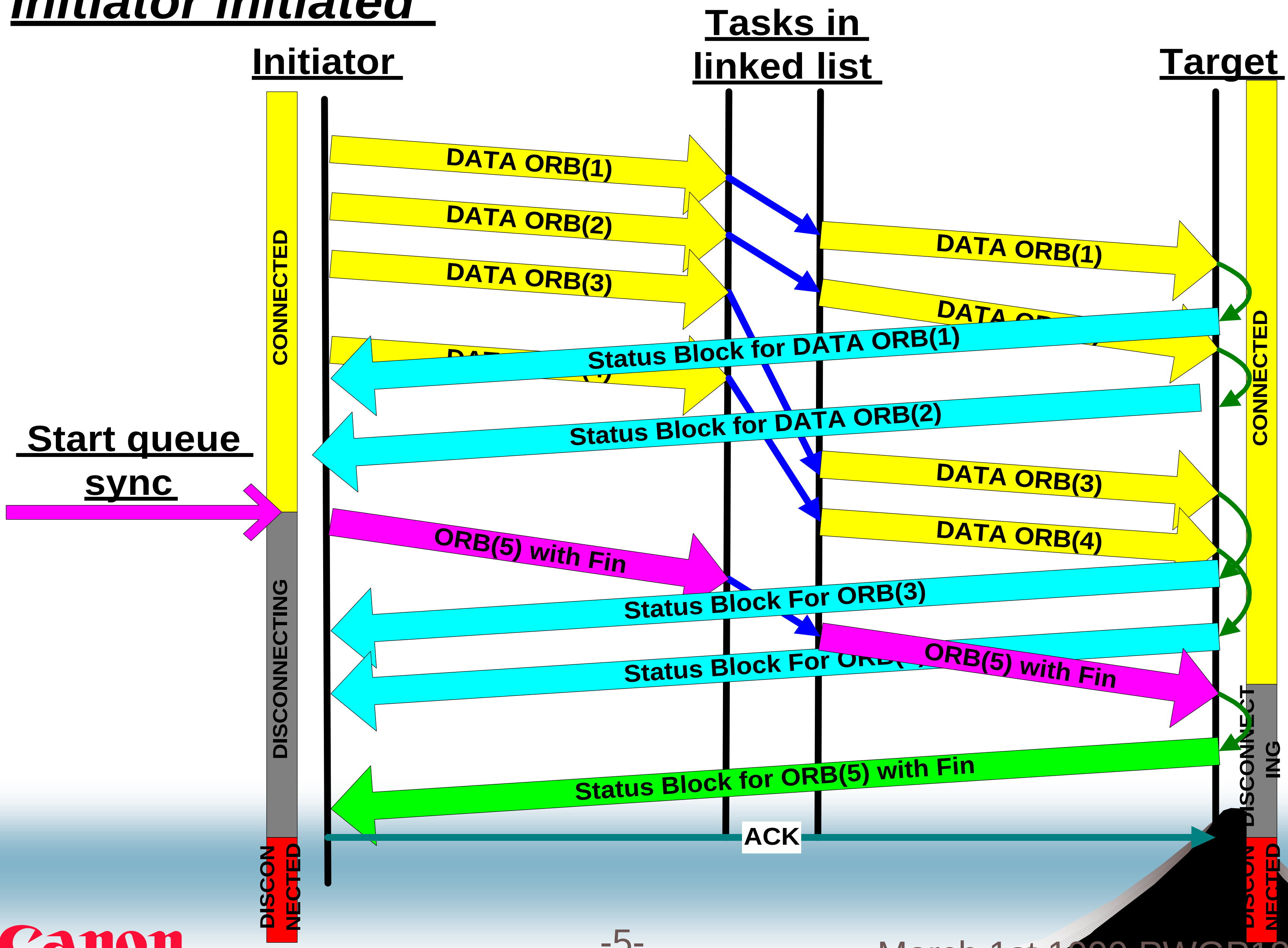
“Disconnect” Procedure

- “Disconnect” consists of ..
  - **Collection of synchronization(s) on each queue**
  - **Release of connection resource**
- Brief “disconnect” procedure example for bi-di case
  - Synchronize both ends of I2T queue and T2I queue respectively on per queue basis using “final bit”.
  - Release connection resource (ex.,Task_slots)

    After every queue in the connection has been synchronized.

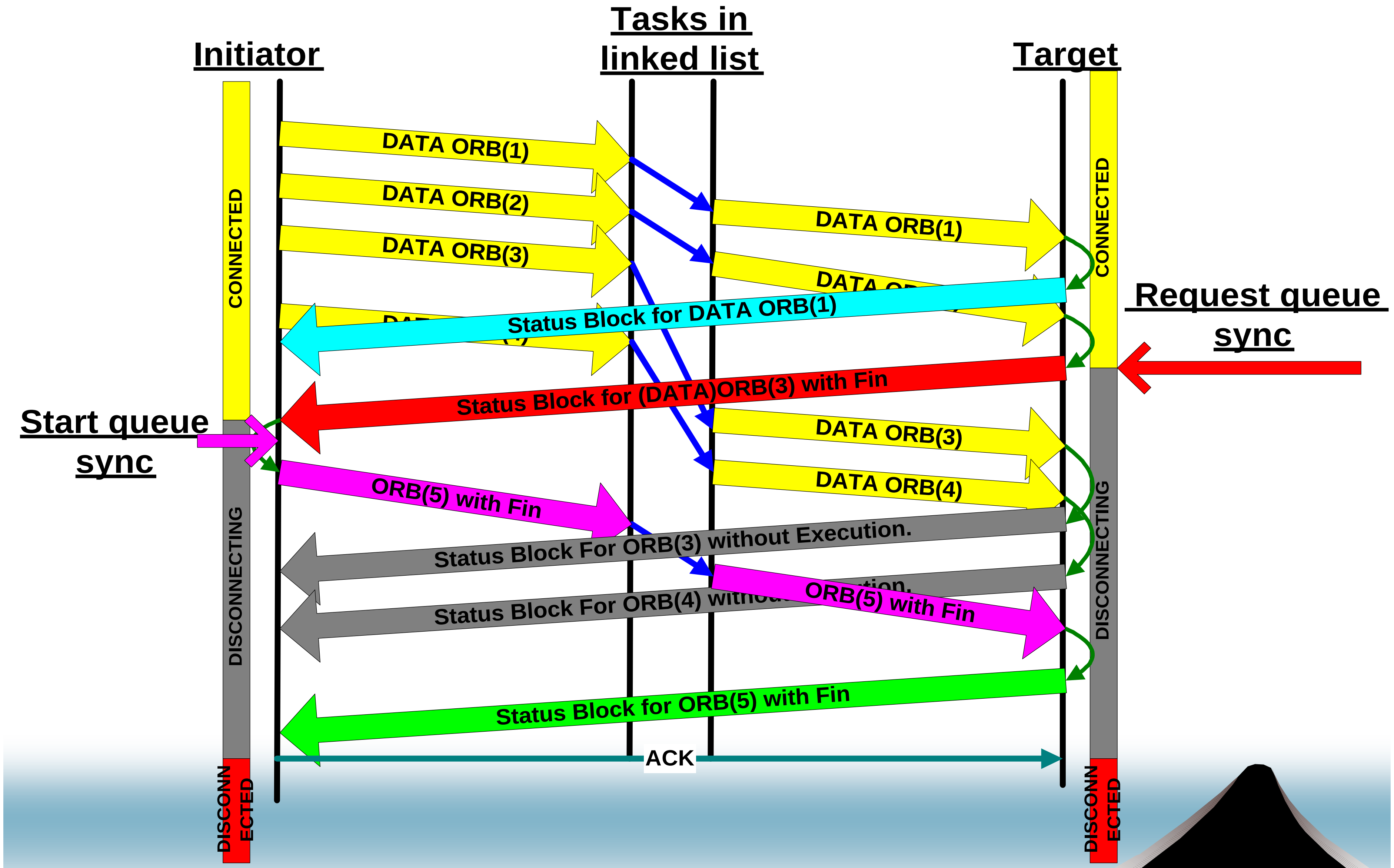
Queue synchronization

Initiator initiated



Queue synchronization

Target initiated

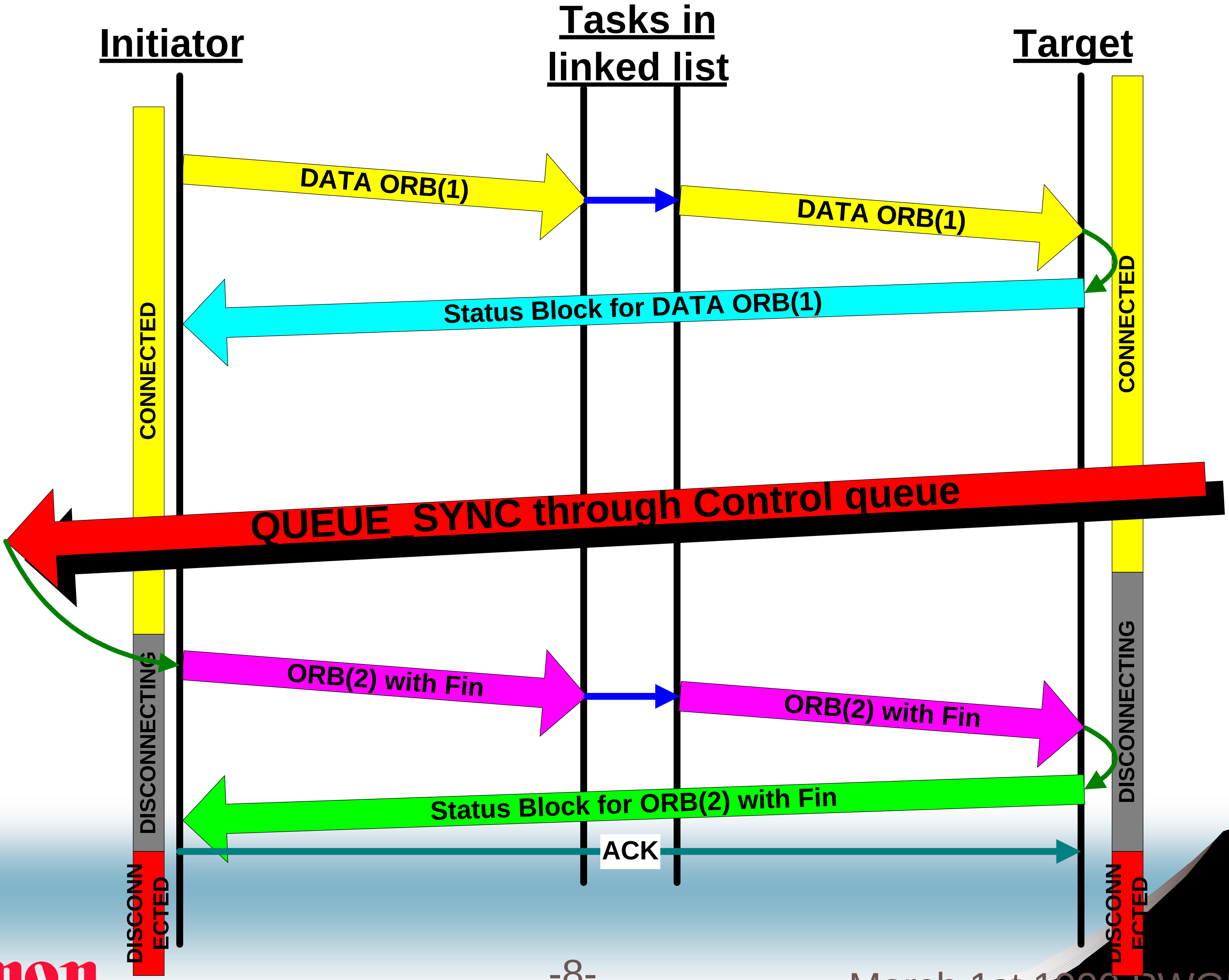


Queue Sync Action

- In case that Target has no Status Block to piggyback queue sync request QUEUE_SYNC action through control queue is used instead.
- QUEUE_SYNC action is used not only by Target but also by Initiator to notify the start of queue sync.
- When Target receives QUEUE_SYNC action Target shall release blocked flow.
- QUEUE_SYNC action does NOT require “reply”.

QUEUE_SYNC

Target initiated (No Status Block)

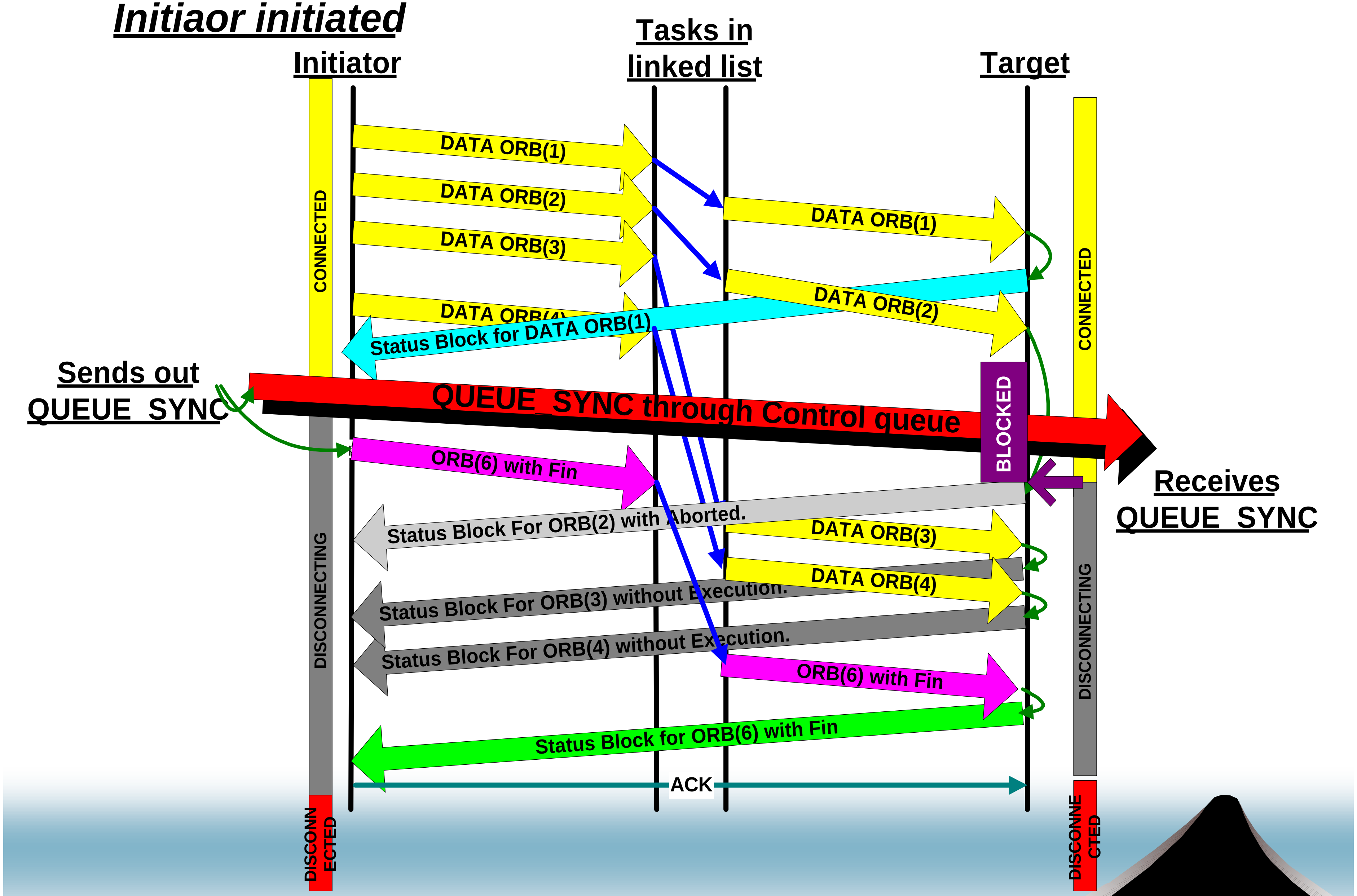


Aborting connection

- Functionality of aborting connection is achieved by the combination of “Final bit” and “QUEUE_SYNC” action.

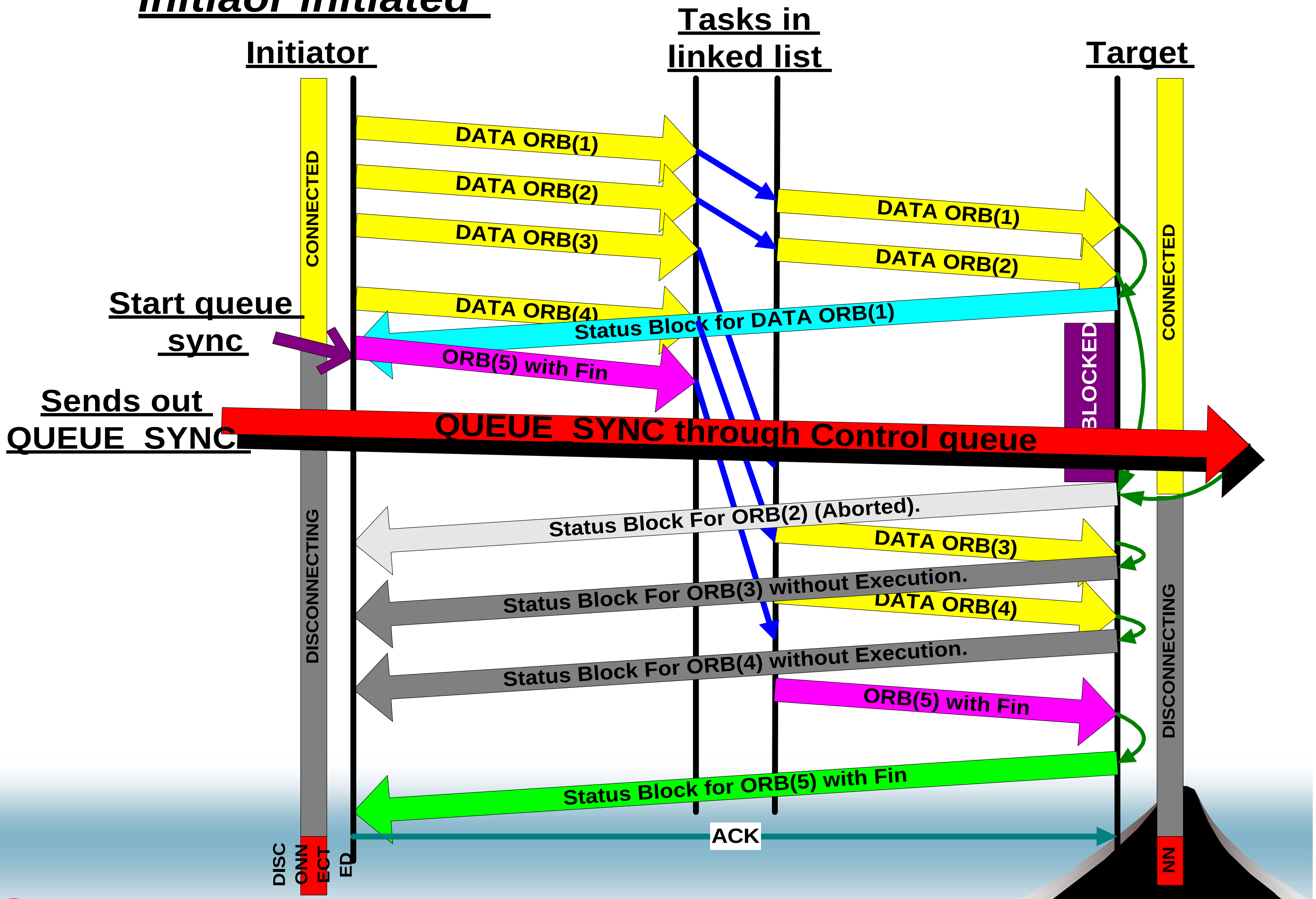
QUEUE_SYNC

Initiaor initiated



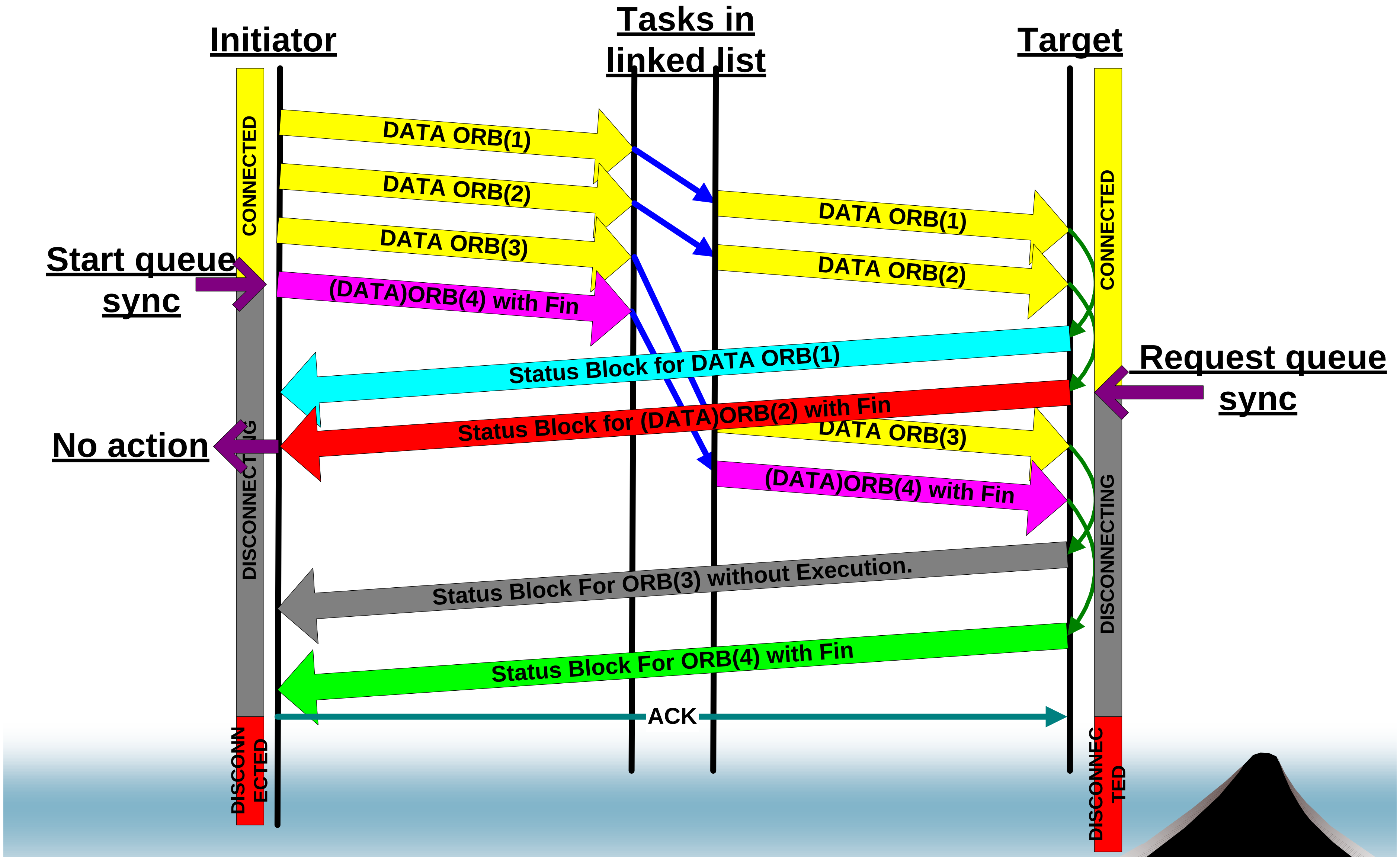
Start queue synchronization but Blocked then QUEUE SYNC

Initiaor initiated



Queue synchronization

Both Initiator and Target initiated at a time(Example)



Conclusion

Disconnect Using “Final Bit” Flag,

- **Prevents disconnect procedure from becoming redundant and unnecessarily complex.**
- **“Disconnect” and “Abort” procedure becomes uniform. Improves testability.**