# How to support Non Blocking mode in our profile?

Takashi Isoda Canon Inc.

# The discussion on “Non-blocking mode” is...

- Start from(before?) New Orleans
- Brian posted  his investigation on this after Philadelphia
- Now still Open

# Background

- Socket API provide two types of operation mode “Blocking” and “Non-Blocking” on a Socket.
- Profile is based on the model that transfers data from or to Initiator’s buffer directly to or from target  (i.e., Shared Memory Model) inherited from SBP-2.

*(TCP follows conventional communication model (buffer to buffer)*

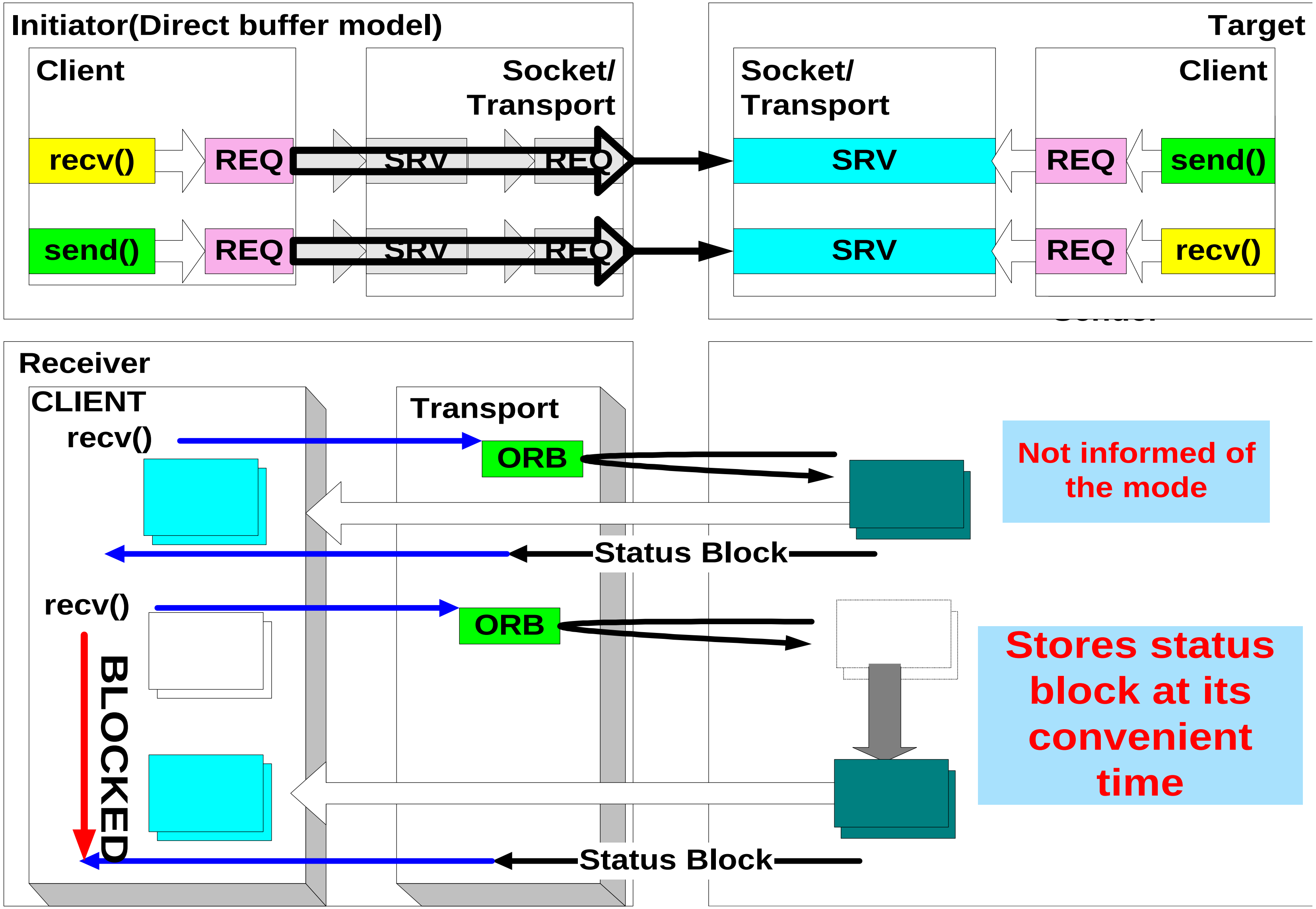
# Problem

- To support both Blocking and Non Blocking modes of “receive()” in the shared memory model, Initiator need to know if the target processing WOULD BLOCK or NOT, and Target need to know the WATERMARK to determine to block ORB or not.

-> **Current profile has a “*target_data_pending*” bit and this may be used to indicate WOULD BLOCK or not, but does not have a parameter to inform the WATERMARK.**

- ***Target may process an ORB differently from Initiator’s intention.***

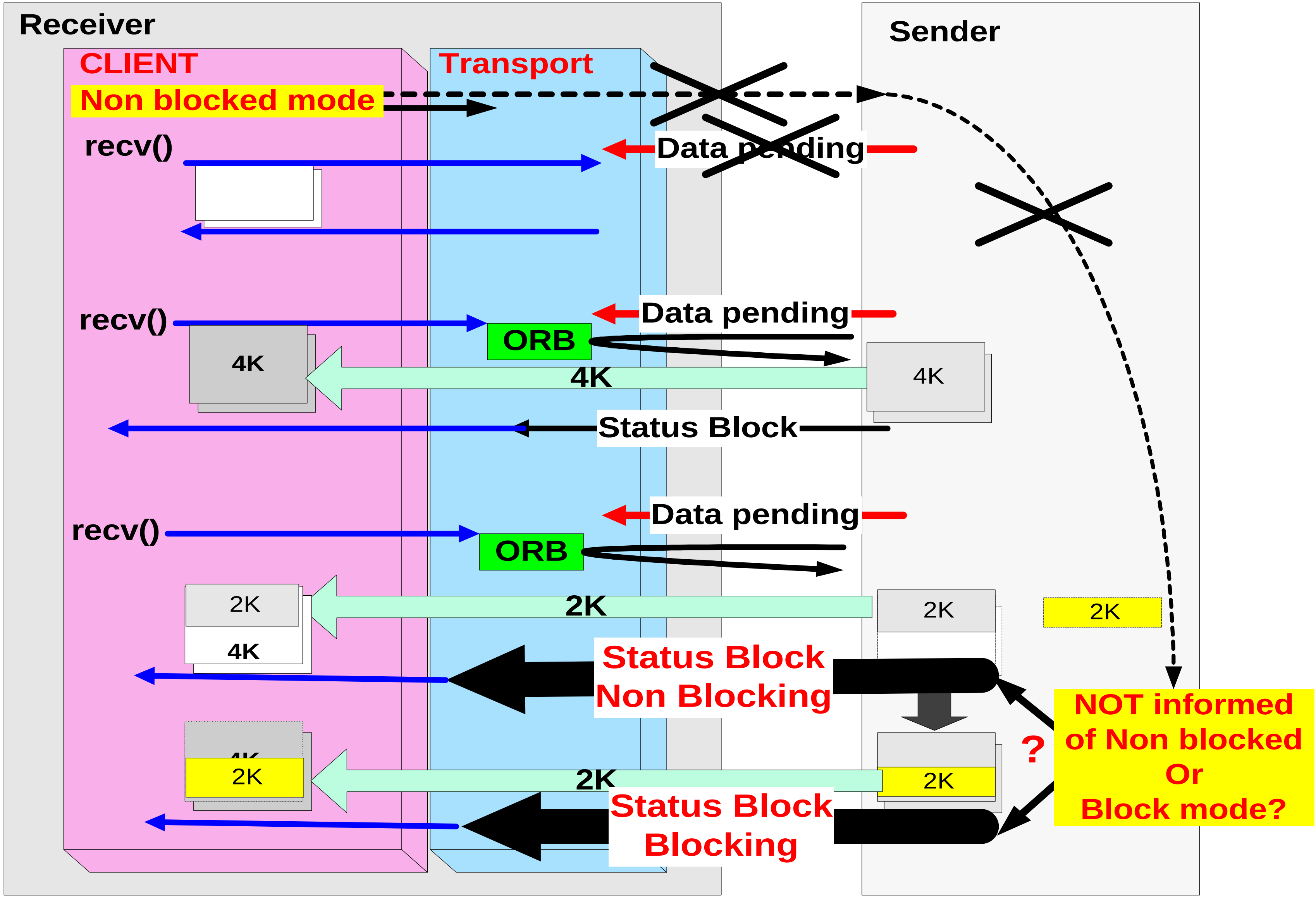
# Current Profile (Direct transfer model)



Not informed of the mode

Stores status block at its convenient time

# To BLOCK or to NON block?, it is question



# Solution?

How about make initiator inform Targert of initiator's socket mode ?...

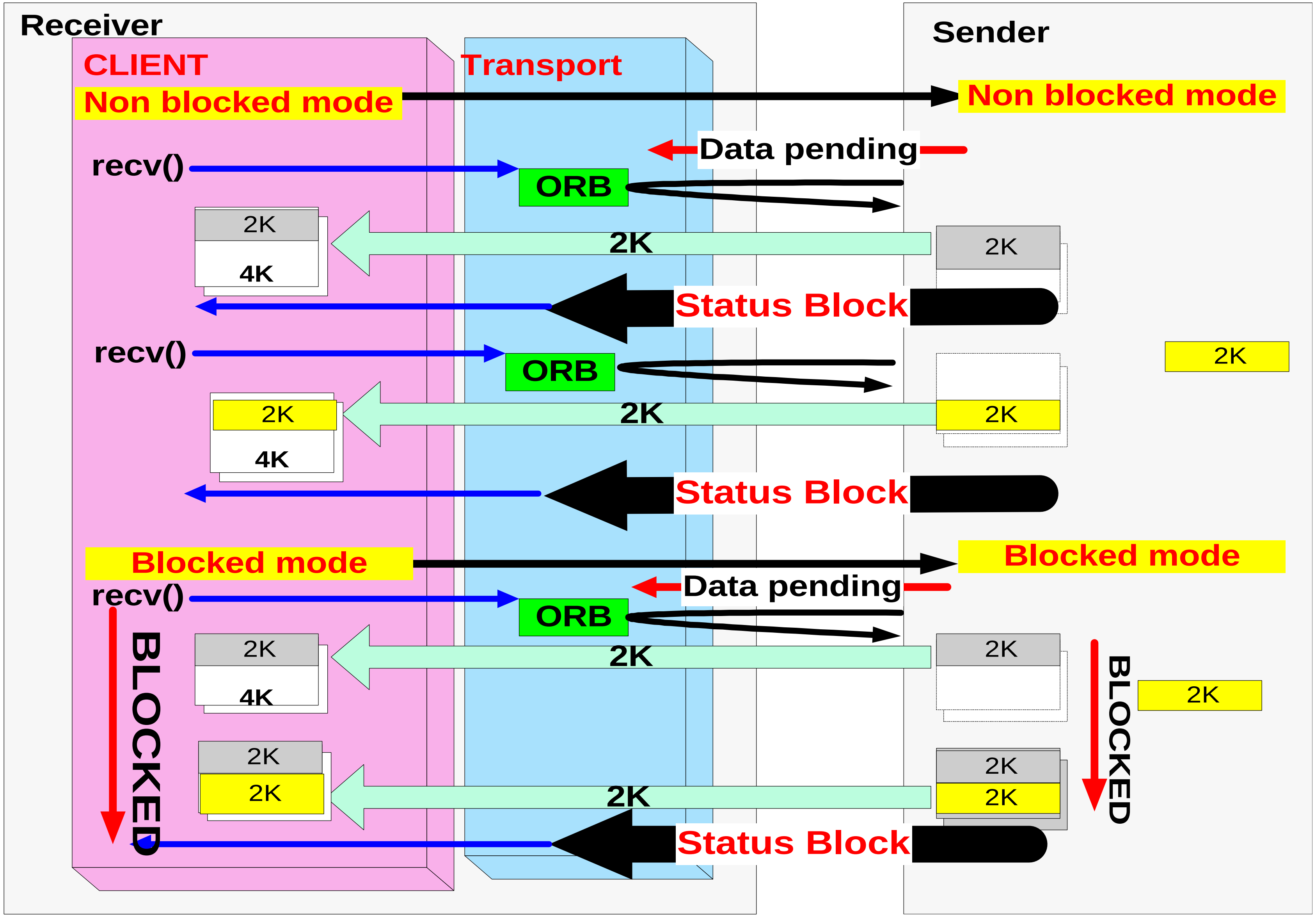
**Add New "Set Transfer mode" Parameter which indicates Blocking or Non blocking**

Initiator can signal this parameter at connect time or via "SET_TRANSFER_MODE" control (additional new control).

*(should signal it when the no active ORB for the queue exists)*

Target shall set the specified queue to specified mode. .

# How it works..


Receiver
CLIENT
Non blocked mode
Transport
Sender
Non blocked mode
Data pending
recv()
ORB
2K
4K
2K
2K
Status Block
recv()
ORB
2K
2K
4K
2K
2K
Status Block
Blocked mode
Blocked mode
Data pending
recv()
ORB
BLOCKED
2K
4K
2K
2K
2K
2K
2K
2K
2K
2K
2K
Status Block
BLOCKED

# Conclusion

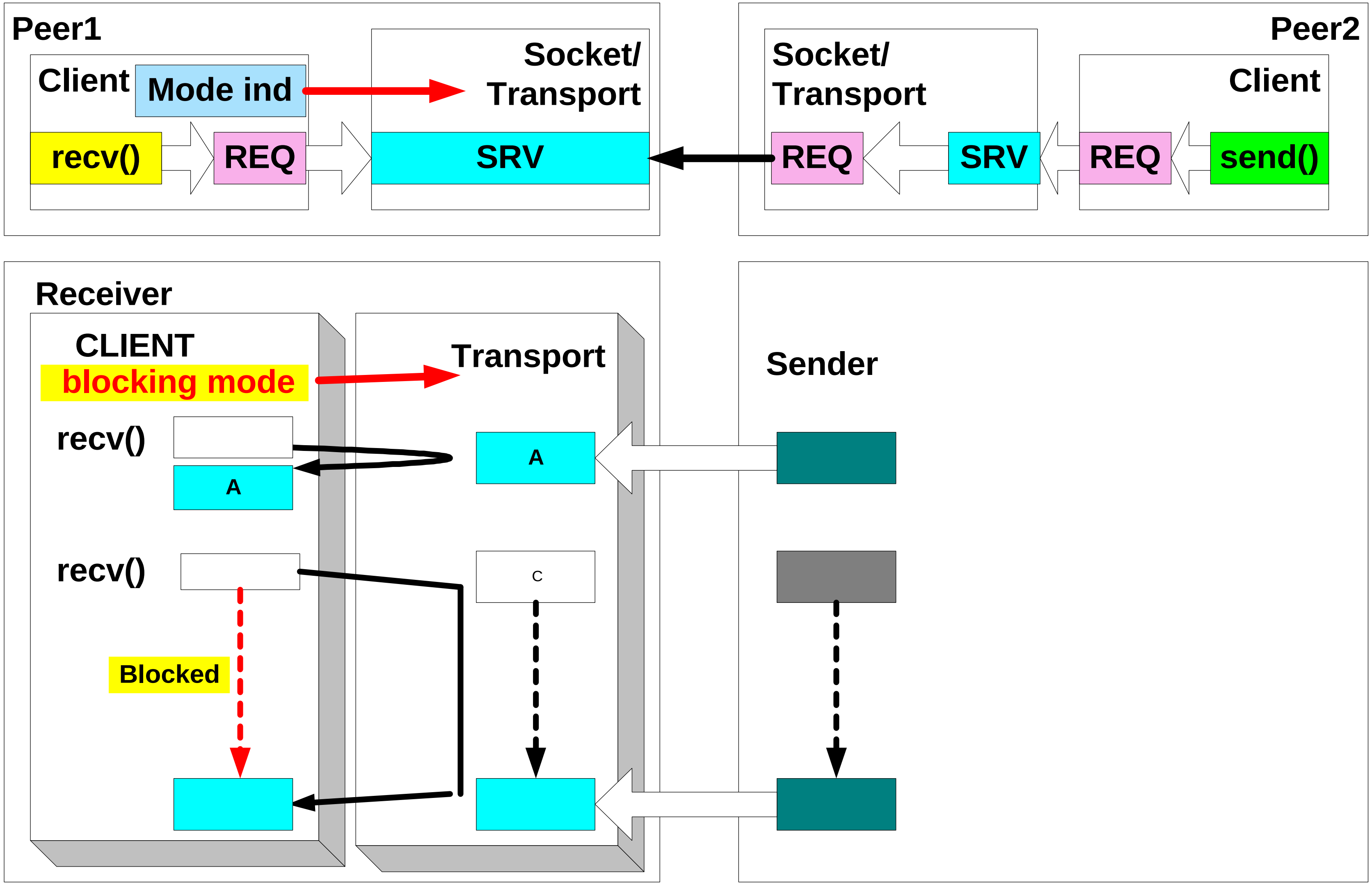
- Is this approach enough to blocking/non-blocking mode on receive()?
- Is there any transport support needed on blocking/non-blocking send()?

-> ***Need more investigation***

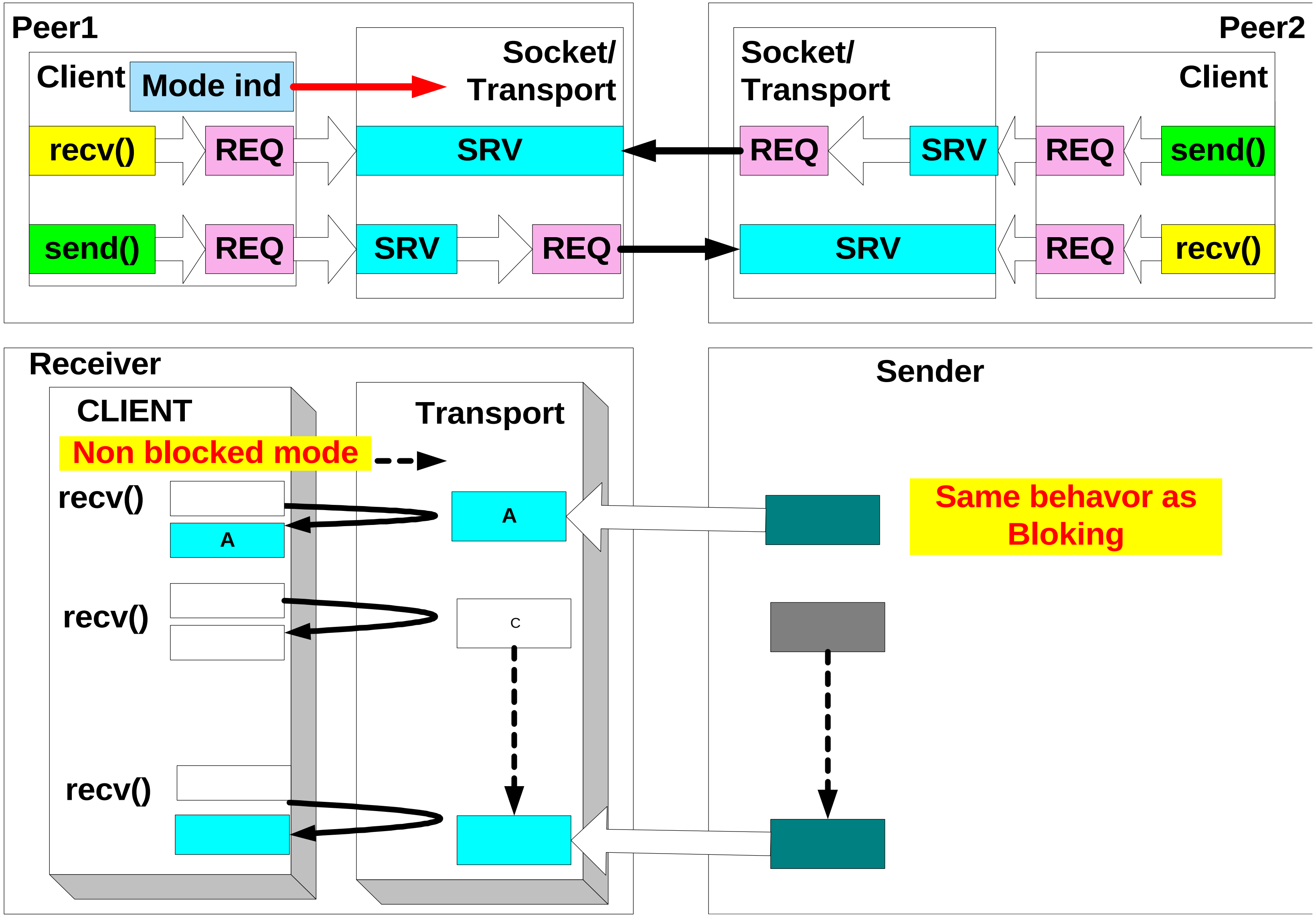
**Ref**.

Why the mode information shall be transferred across the wire though it shall not be in case of Socket on TCP.

# Socket on TCP(Blocking)

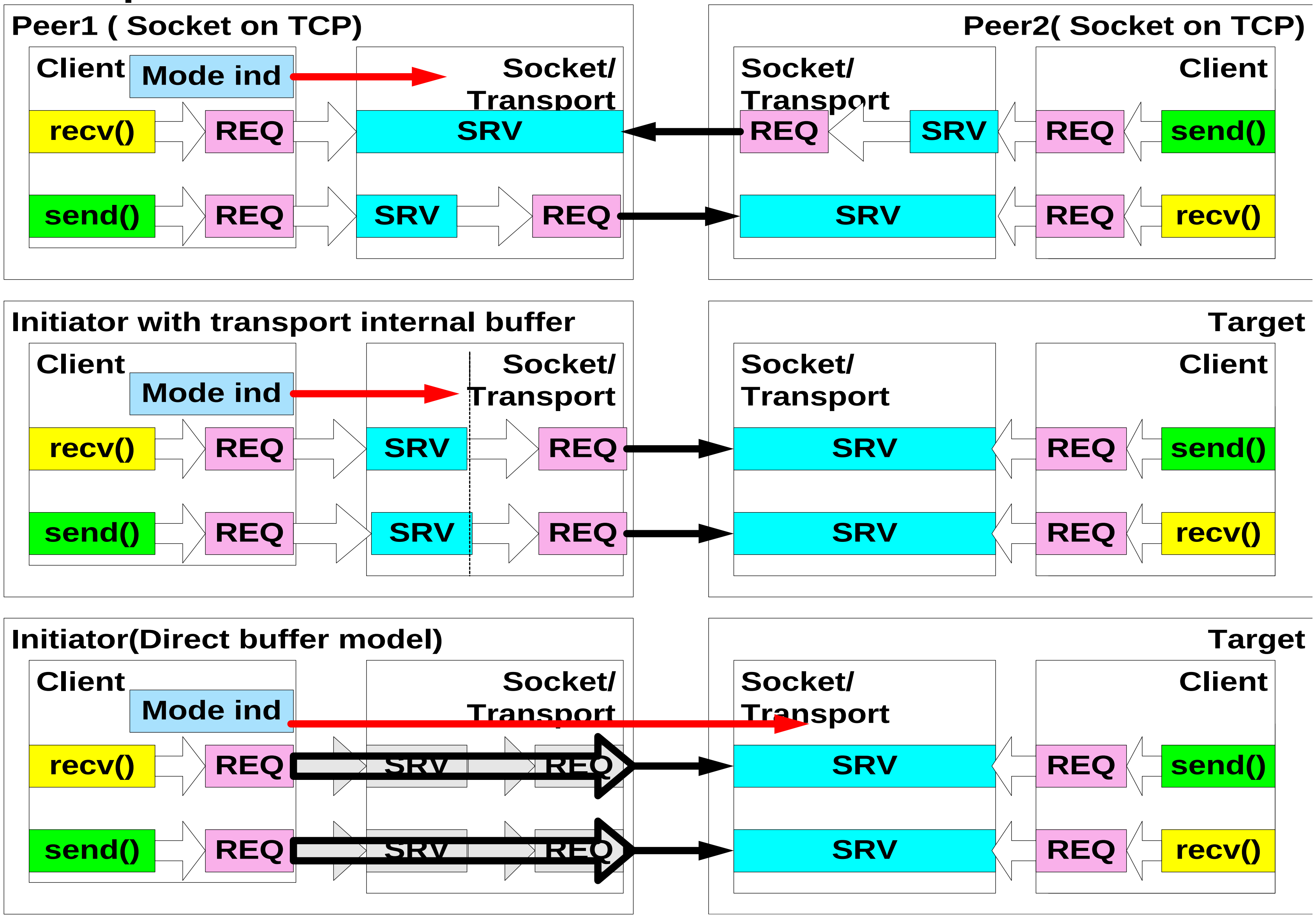


# Socket on TCP(Non Blocking)



Same behavor as Bloking

# Comparison of the models


Peer1 ( Socket on TCP)
Client
Mode ind
Socket/ Transport
recv()
REQ
SRV
send()
REQ
SRV
REQ
Peer2( Socket on TCP)
Socket/ Transport
Client
REQ
SRV
REQ
send()
SRV
REQ
recv()
Initiator with transport internal buffer
Client
Mode ind
Socket/ Transport
recv()
REQ
SRV
REQ
send()
REQ
SRV
REQ
Target
Socket/ Transport
Client
SRV
REQ
send()
SRV
REQ
recv()
Initiator(Direct buffer model)
Client
Mode ind
Socket/ Transport
recv()
REQ
SRV
REQ
send()
REQ
SRV
REQ
Target
Socket/ Transport
Client
SRV
REQ
send()
SRV
REQ
recv()