Isochronous owner observation

Proposal for March 19 and 20 P1394.1 working group

Kazunobu Toguchi

Media Processing Laboratories Sony Japan

togu@av.crl.sony.co.jp

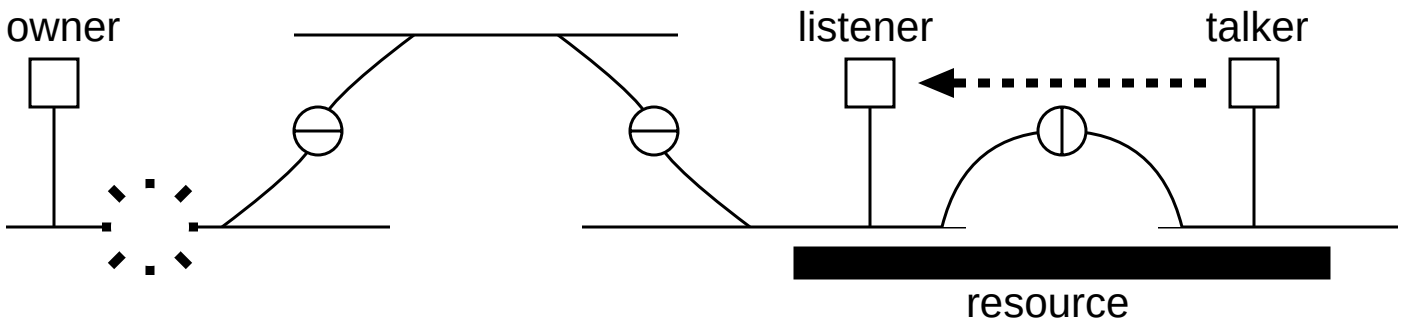
Agenda

1. Problem; isochronous resource can’t be released
2. Requirements
   - Isochronous owner observation
   - To release resource instead of owner
3. Isochronous owner observation
   - Observer definition
   - Observation method
   - Registers
   - Observation data flow
   - Observer candidates
   - Observer initialization
4. Conclusion

Problem

If isochronous owner is disconnected from net by accident, no node can release the resources that the owner reserved before.

There are not any nodes which should investigate if owner exists in net or not under the current draft.



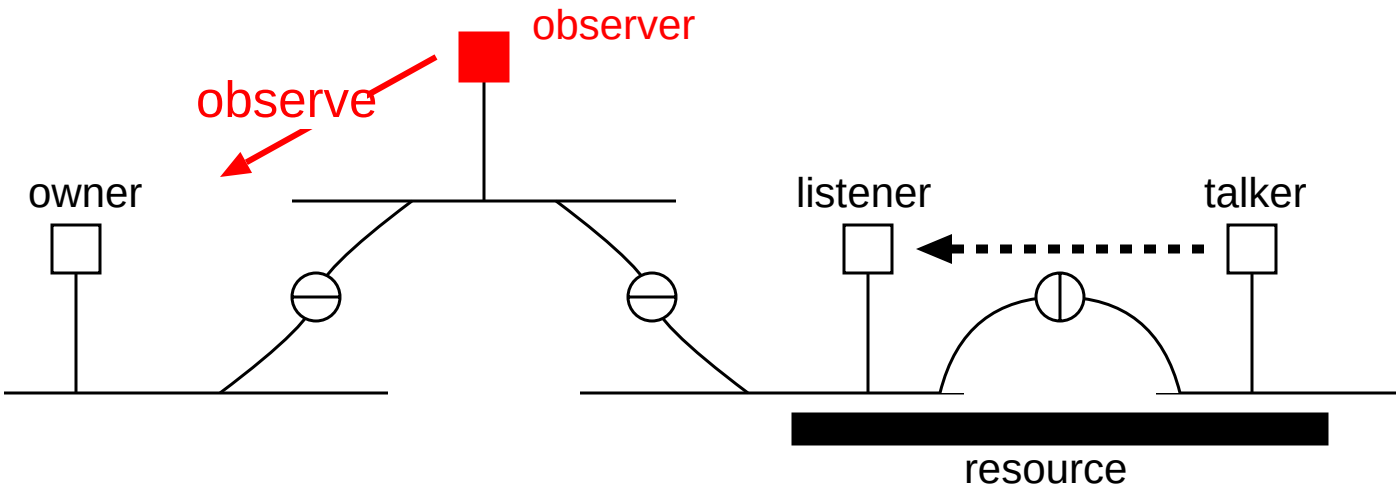
Requirements

We have two requirements of the problem;

1. net should detect the accident that owner disappeared,
2. and release resources instead of owner.

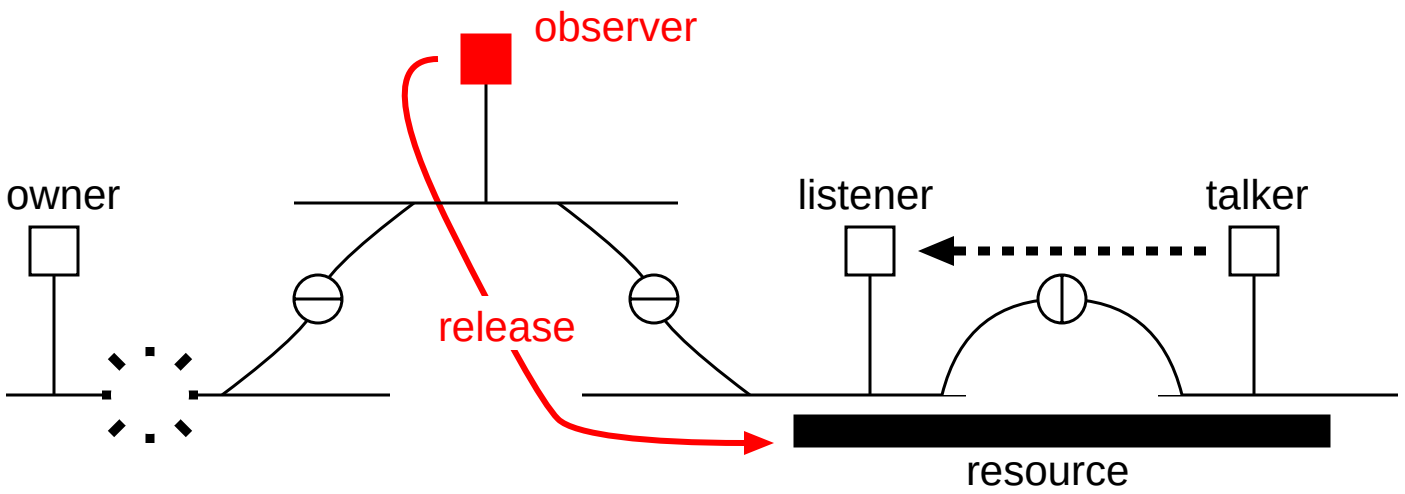
Definition of observer (1 / 2)

We define the node which should observe
isochronous owner as ’observer.’
Observer should investigate whether isochronous owner exists in net.



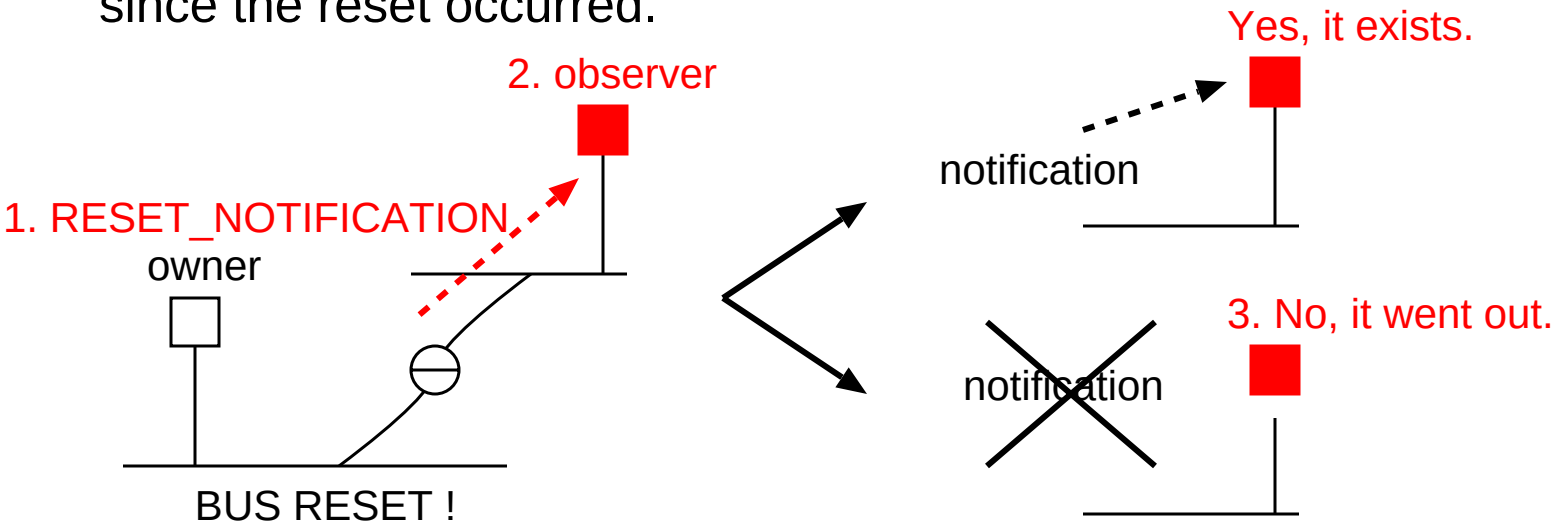
Definition of observer (2 / 2)

If observer knows that owner disappeared,
it might have to release the resources which the owner reserved for isochronous transaction before instead of the owner.



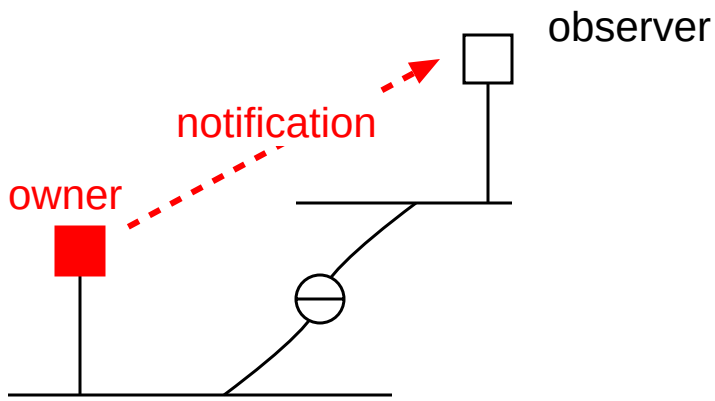
Investigation method (1 / 2)

1. When observer gets RESET_NOTIFICATION,
it should check if the reset occurred at owner's local bus.
2. If it occurred, the observer should wait for notification
from the owner.
3. Then, If the observer can get notification from the owner,
the observer knows that the owner has still stayed in the net.
4. If not, the observer knows that the owner has already disappeared
since the reset occurred.



Investigation method (2 / 2)

The owner whose local bus had bus reset should send notification to observer.



Registers (1 / 3)

We define two register fields for isochronous owner observation.

1. owner_status field

Observer should have this field.
It should specify state of owner, as encoded by the table below.

owner_status	owner's existence
0	unconfirmed
1	existence

The observer which knows that owner's local bus had bus reset should update owner_status field with zero.
The owner whose local bus had bus reset should update owner_status field with one so that observer can know its existence.

Registers (2 / 3)

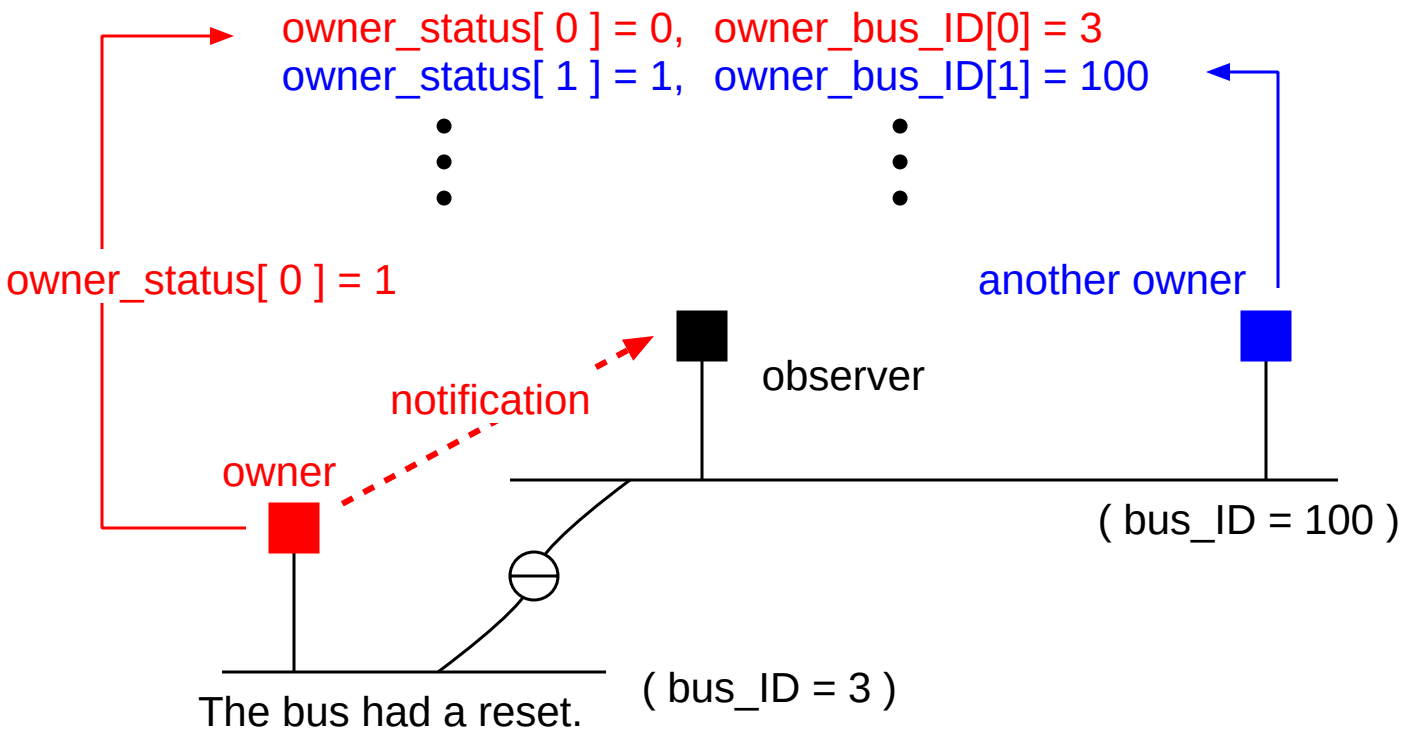
2. owner_bus_ID field

Observer should have this field.
This field should specify owner's bus_ID.
owner_bus_ID field enables observer to distinguish RESET_NOTIFICATION.

```
if ( RESET_NOTIFICATION.bus_ID == owner_bus_ID ) {

   /*  Bus reset occurred at owner's local bus.                  */
   /*  Thus, owner should wait for notification from owner. */

} else {

   /* Observer ignores the RESET_NOTIFICATION. */

}
```

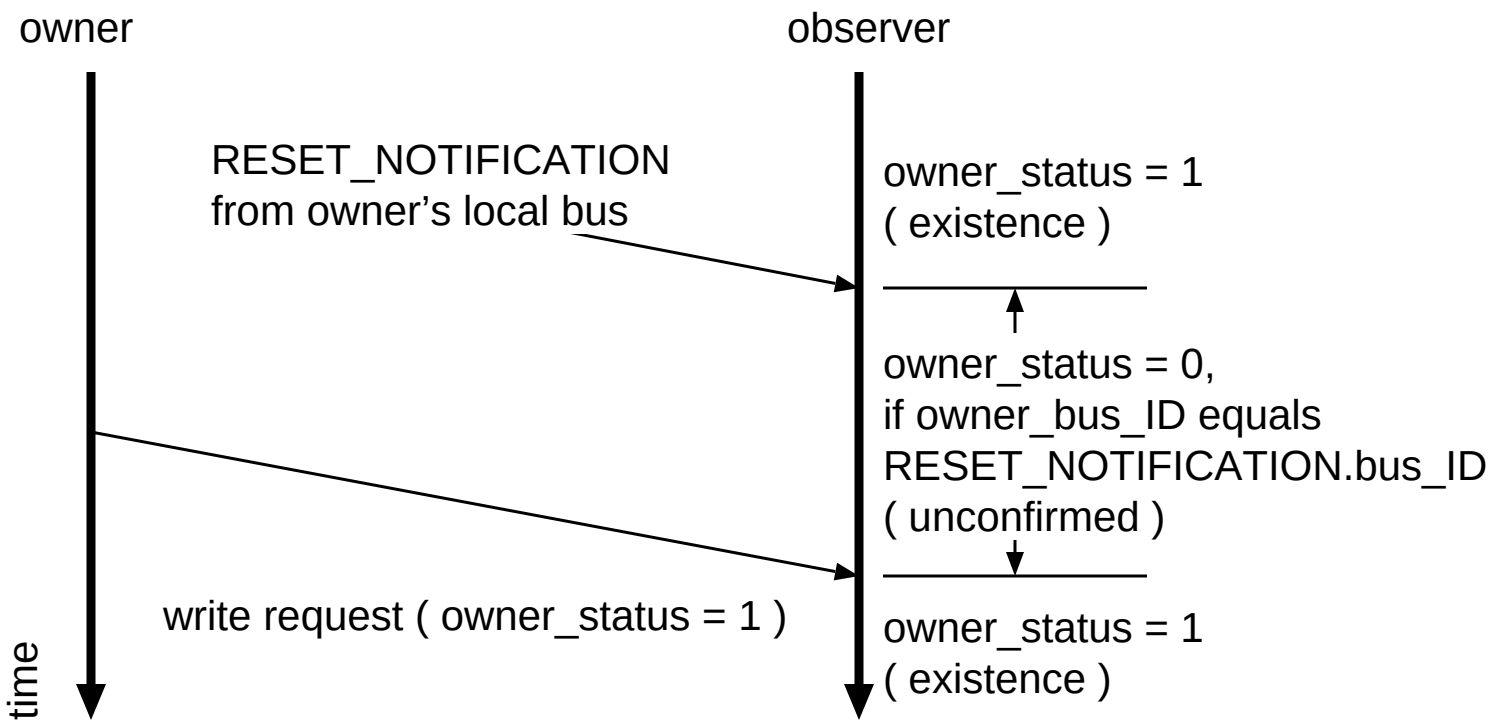
Registers (3 / 3)

To enable observer to investigate as many owners as possible, observer should have some pairs of owner_status and owner_bus_ID fields in array of registers, such as STREAM_CONTROL registers.



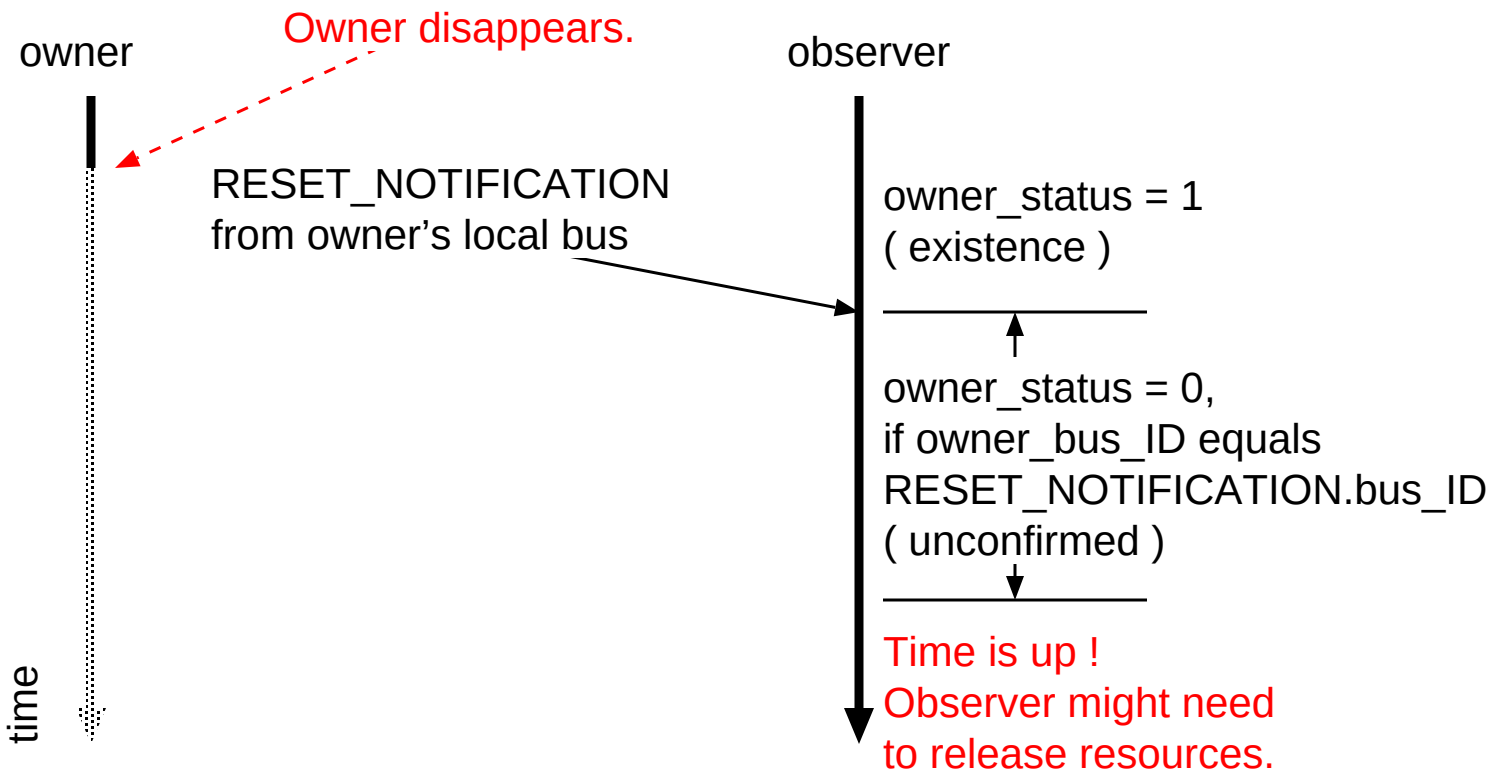
Observation data flow (1 / 2)

If owner still exists after a bus reset,
the owner should update the owner_status filed with
the value of one in order to let observer know its existence.



Observation data flow (2 / 2)

Observer misses notification from owner,
then observer knows that owner is disconnected from net.

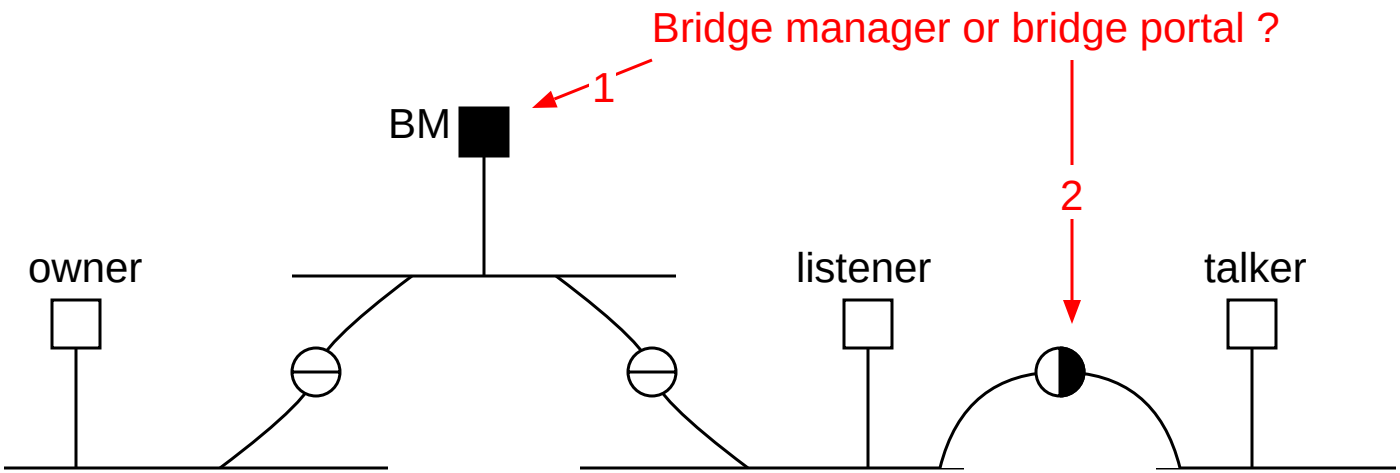


Observer candidates

We have two candidates for observer:

1. Bridge portal

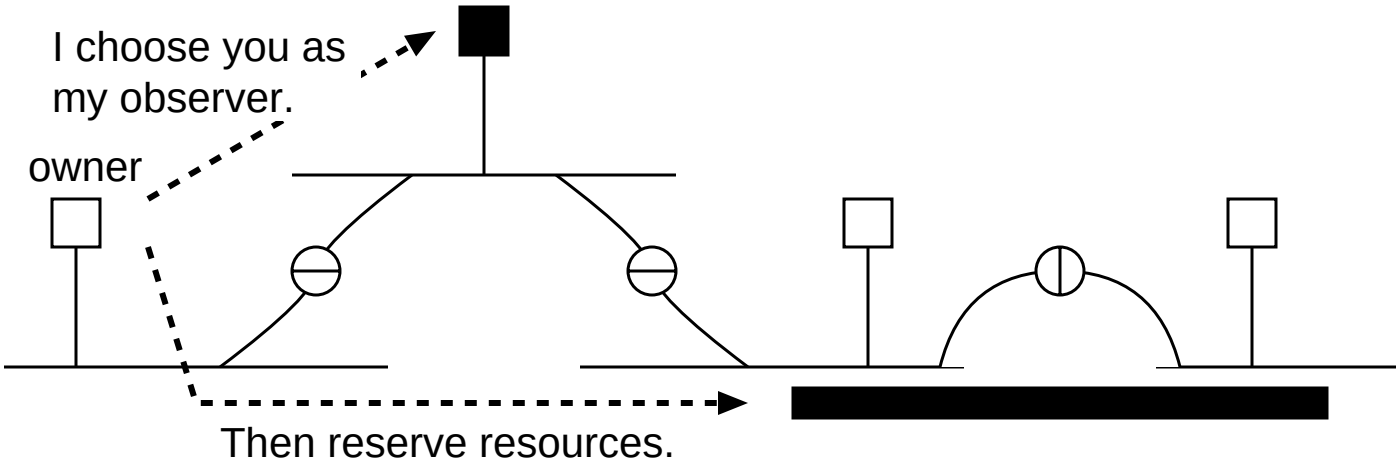
2. Bridge manager



Observer initialization

Before resource reservation,
isochronous owner should set observer node,
in case owner disappears while resource reservation.

Owner should update both owner_status and
owner_bus_ID fields, then reserve resources for transaction.



Conclusion

1. Observer investigates if owner exists in net
2. Even if owner is disconnected from net by accident, owner can know the accident
3. Thus observer or another can start to release resources

To be considered

1. Whether observer should release resources ?
2. Which candidate, bridge manager or portal, should become observer ?