It finally could work.. Actually I get a problem here and it's confusing. My first attempt on the **Send** is to change it into **entry,** the code is above:

*entry Send (To : Box; Data : Message) when*

*(for some B in Box => not Store (B).Available) is*

*begin*

*if Store (To).Available then*

*requeue Send;*

*else*

*Store (To) := (Available => True, Data => Data);*

*end if;*

*end Send;*

As a result, the **Broadcast** and **Multicast** are struggling with the unterminated loop. In other words, when I debugged the procedure, it seemed that **Broadcast** and **Multicast** started to call themselves after finished one turn, which leaded to the result:

*send*

*send*

*Task 1 received: CYAN*

*Task 2 received: CYAN*

*Task 1 received: CYAN*

*Task 2 received: CYAN*

*Task 3 received: CYAN*

*send*

*Task 3 received: CYAN*

*send*

*Task 2 received: CYAN*

*Task 3 received: CYAN*

*Task 1 received: CYAN*

Presumably there are some recall problems with **entry Send** which is called in **Broadcast** and **Multicast,** because it operated normally once I changed **Send** back into procedure and tested it.