
MODULE *P2PBroadcastSpec*

Spec for a reliable broadcast. This captures the requirement that if any processor sends a message then eventually all other processes receive the message.

EXTENDS *Naturals*, *Sequences*

CONSTANT

Proc, Set of processes

Data

VARIABLES

sent, All messages sent

received All messages received

vars $\triangleq \langle sent, received \rangle$

Message $\triangleq [from : Proc, data : Data]$

Init $\triangleq \wedge sent = \langle \rangle$
 $\wedge received = [m \in Message \mapsto \{\}]$

TypeOK $\triangleq \wedge sent \in Seq(Message)$
 $\wedge received \in [Message \rightarrow SUBSET Proc]$

Send $\triangleq \wedge \exists m \in Message : sent' = Append(sent, m)$ Always enabled
 $\wedge UNCHANGED \langle received \rangle$

Recv(*m*) $\triangleq \wedge sent \neq \langle \rangle$
 $\wedge m = Head(sent)$
 $\wedge sent' = Tail(sent)$
 $\wedge \exists p \in Proc : received' = [received \text{ EXCEPT } ![m] = @ \cup \{p\}]$

Next $\triangleq \exists m \in Message : Send \vee Recv(m)$

Spec $\triangleq Init \wedge \Box [Next]_{vars}$

FairSpec is *Spec* with the addition requirement that it keeps taking steps.

FairSpec $\triangleq Spec \wedge WF_{vars}(Next)$

* Modification History
* Last modified Wed Apr 05 12:19:39 CEST 2023 by kulpreet
* Created Wed Apr 05 09:47:12 CEST 2023 by kulpreet