

Assertional proof of sliding window protocol over lossy channel

The stuff in the boxes is what you were supposed to give me.

Here are the predicates we already have:

$$A_1 : \text{sbuff.domain} = [0..ng - na - 1]$$

$$A_2 : 0 \leq na \leq ns \leq ng$$

$$A_3 : \text{rbuff.domain} = [0..RW - 1]$$

$$A_4 : 0 \leq nd \leq nr \leq nd + RW$$

$$B_2 : (nd = \text{Fxy.rhx.size}) \text{ and } (ng = \text{Fxy.txh.size})$$

$$B_3 : [\text{forall } 0..RW-1 \ i :: (\text{rbuff}[i] \text{ is empty}) \text{ or } (\text{rbuff}[i] = \text{Fxy.txh}[nd+i])]$$

$$B_4 : [\text{forall } 0..RW-1 \ i, nd+i < nr :: \text{rbuff}[j] \text{ is non-empty}]$$

$$B_5 : [\text{forall } 0..RW-1 \ i, nd+i \geq ns :: \text{rbuff}[j] \text{ is empty}]$$

$$B_6 : na \leq nr \leq ns \leq na + SW$$

$$C_1 : \text{data message } j \text{ in transit} \Rightarrow nr - N + RW \leq j \leq nr + N - 1$$

$$C_2 : \text{ack message } j \text{ in transit} \Rightarrow ns - N + 1 \leq j \leq na + N$$

Recall that C_1 and C_2 are not really defined until we can express “msg j is in transit in L_{xy} ” in terms of $L_{xy}.txh$ and $L_{xy}.rxh$. Similarly with L_{yx} . This is achieved by defining the following predicates.

$L_{xy}.transit(m) : L_{xy}.rxh \circ \langle m \rangle$ subsequence of $L_{xy}.txh$

$L_{yx}.transit(m) : L_{yx}.rxh \circ \langle m \rangle$ subsequence of $L_{yx}.txh$

Thus “data j in transit in L_{xy} ” is formally denoted by $L_{xy}.transit(j)$. And “data j in transit behind data k in L_{xy} ” is formally denoted by $L_{xy}.transit(k, j)$.

So C_1 and C_2 are rewritten as follows:

$$C_1 : L_{xy}.transit(j) \Rightarrow nr - N + RW \leq j \leq nr + N - 1$$

$$C_2 : L_{yx}.transit(j) \Rightarrow ns - N + 1 \leq j \leq na + N$$

The following hold (wrt the composite system Z):

- Predicates A_1, A_2, A_3, A_4, B_2 individually satisfy the invariance rule (i.e., for $X = A_1, A_2, A_3, A_4, B_2$, $\{\text{true}\}init()\{X\}$ holds, and for every event e : $\{X\}init()\{X\}$ holds).
- $B_{3..6}$ satisfies the invariance rule assuming $C_{1,2}$ (i.e., $\{\text{true}\}init()\{B_{3..6}\}$ holds, and for every event e : $\{B_{3..6}, C_{1,2}\}init()\{B_{3..6}\}$ holds).

So what remains is to establish $\square C_{1,2}$. Define the following predicates.

$$D_1 : L_{xy}.transit(j) \Rightarrow j \leq ns - 1$$

$$D_2 : L_{xy}.transit(j) \Rightarrow j \geq nr - SW$$

$$D_3 : (L_{xy}.transit(j) \text{ and } \text{rbuff}[i] \neq \text{empty}) \Rightarrow j \geq nd + i - SW + 1$$

$$D_4 : L_{xy}.transit(k, j) \Rightarrow j \geq k - SW + 1$$

$(D_1 \wedge D_2 \wedge B_6) \Rightarrow C_1$ holds. $D_{1..4}$ satisfies invariance rule assuming $B_{1..6}$. Some details for the latter claim:

- D_1 satisfies invariance rule given B_6 .

[Some more details: D_1 can only be falsified by sending a data message, and $\{D_1, B_6\}$ “send data msg” $\{D_1\}$ holds.]

- D_2 satisfies invariance rule given B_6, D_3 .

[Some more details: B_6 ensures that data send satisfies D_2 ; D_3 ensures that data reception satisfies D_2 .]

- D_3 satisfies invariance rule given B_5, B_6, D_3, D_4 .

[Some more details: D_3, B_5, B_6 ensures that data send satisfies D_3 ; D_4 ensures that data reception satisfies D_3 .]

- D_4 satisfies invariance rule given B_6, D_1, D_2, D_4 .

[Some more details: D_2, D_1, B_6 ensures that data send satisfies D_4]

So all that remains is to establish $\square C_2$. Define the following predicates.

$E_1 : \text{Lyx.transit}(j) \Rightarrow j \geq na$

$E_2 : \text{Lyx.transit}(j, k) \Rightarrow k \geq j$

$(E_1 \wedge E_2 \wedge B_6) \Rightarrow C_2$ holds. $E_{1,2}$ satisfies invariance rule assuming $B_{1..6}$.

In summary, $A_{1..4} \wedge B_{2..6} \wedge D_{1..4} \wedge E_{1,2}$ satisfies the invariance rule.