Message Passing Case-study (Time Added Manually)

Table of Contents

Context c00

CONTEXT
c00
SETS
NET_MESSAGE
CONSTANTS
Data
ACK
FIN
Send_Message
last // *Last sequence number of sendable message*
DataTDL // *Deadline duration to transfer Send_Message*
AP_DATA // *Application Data*
AXIOMS
axm1 : partition(NET_MESSAGE, Data, {ACK}, {FIN})
axm2 : AP_DATA ⊆ Data
axm3 : Send_Message ∈ ℕ ⇸ NET_MESSAGE
axm4 : ran(Send_Message) = Data ∪ {FIN}
axm5 : finite(Send_Message) ∧ Send_Message ≠ ∅
axm6 : last = card(Send_Message)−1
axm7 : ∀ n· n ∈ dom(Send_Message) ⇒ 0 · · n ⊆ dom(Send_Message)
axm8 : dom(Send_Message)= 0 · · last
axm9 : ∀ m · m ∈ dom(Send_Message) ∧ m ≠ last ⇒ Send_Message(m) ≠ FIN
axm10 : Send_Message(last) = FIN
axm11 : ∀ n· n ∈ ℕ ∧ n < last ⇒ Send_Message(n) ∈ Data
axm12 : DataTDL ∈ ℕ
axm13 : DataTDL > 0
axm14 : ∀ m,x,y · m ∈ Send_Message ∧ m=x↦y ∧ y = FIN ⇒ x = last
axm15 : ∀ x · x ∈ dom(Send_Message) ⇒ x ∈ 0 · · last
axm16 : ∀ x · x ⊆ ℕ ∧ finite(x)∧ x≠ ∅ ⇒ max(x)+1 ∉ x
END

Context c01

CONTEXT
c01
EXTENDS
c00
CONSTANTS
PacketTDL
AXIOMS
axm1 : PacketTDL ∈ ℕ
axm2 : DataTDL ≥ (last+1) ∗ PacketTDL
END

Context c02

CONTEXT
c02
EXTENDS
c01
CONSTANTS
MaxResend
ResendingDL
AXIOMS
axm1 : MaxResend > 1
axm5 : ResendingDL > 0
axm6 : PacketTDL ≥ (MaxResend+1)∗ResendingDL
END

Context c03

CONTEXT

c03

EXTENDS

c02

CONSTANTS

ChannelDL
ReceiverDL

AXIOMS

axm1 : ChannelDL ∈ ℕ
axm2 : ResendingDL > 2∗ChannelDL
axm3 : ReceiverDL ∈ ℕ
axm4 : ReceiverDL ≥ MaxResend ∗ ResendingDL+2∗ChannelDL
axm5 : ReceiverDL ≥ PacketTDL

END

Machine m00

MACHINE

m00 // *Deadline(Start_Transfering, Transfering ∨ Error, DataTDL)*

SEES

c00

VARIABLES

transfered0 // *Whole transfered Message*
Transfered // *Flag*
Start_Transfering // *Flag*
Error // *Flag*
time
Start_TransferingT

INVARIANTS

inv1 : transfered0 ⊆ Send_Message
inv2 : Start_Transfering ∈ BOOL
inv3 : Transfered ∈ BOOL
inv4 : Error ∈ BOOL
inv5 : time∈ℕ
inv6 : Start_TransferingT∈ ℕ
inv7 : dom(Send_Message) = 0 · · last
inv8 : last ≥ 0

EVENTS

INITIALISATION ≙
STATUS
ordinary

BEGIN
act1 : transfered0 ≔ ∅
act2 : Transfered ≔ FALSE
act3 : Start_Transfering ≔ FALSE
act4 : Error ≔ FALSE
act5 : time ≔ 0
act6 : Start_TransferingT ≔ 0
END

Start_Transfering ≙
STATUS
ordinary

WHEN
grd1 : Start_Transfering = FALSE
THEN
act1 : Start_Transfering ≔ TRUE
act2 : Start_TransferingT ≔ time
END

Transfered ≙
STATUS
ordinary

WHEN
grd1 : Transfered = FALSE
grd2 : Start_Transfering = TRUE
grd3 : Error = FALSE
THEN
act1 : Transfered ≔ TRUE
act2 : transfered0 ≔ Send_Message
END

Error ≙
STATUS
ordinary

WHEN
grd1 : Start_Transfering = TRUE
grd2 : Transfered = FALSE
grd3 : Error = FALSE
THEN
act1 : Error ≔ TRUE

END

Tick_Tock ≙
STATUS
ordinary
ANY
tick
WHERE
grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : Start_Transfering=TRUE∧Transfered = FALSE ∧ Error=FALSE⇒time+tick≤Start_TransferingT+DataTDL
THEN
act1 : time ≔ time+tick
END

END

Machine m01

MACHINE
m01 // *Deadline(Start_Transfering, Transfering(last)∨Error, DataTDL)*
REFINES
m00
SEES
c00
VARIABLES
transfered // *Whole transfered Message*
Transfering // *Flag*
Error // *Flag*
time
TransferingT
Start_Transfering
Start_TransferingT
INVARIANTS
inv1 : transfered ⊆ Send_Message
inv2 : Start_Transfering ∈ BOOL
inv3 : Transfering ⊆ dom(Send_Message)
inv4 : Error ∈ BOOL
inv5 : time∈ℕ
inv6 : TransferingT∈ Transfering ⟶ ℕ
inv7 : Start_TransferingT∈ ℕ
inv8 : ∀ n· n ∈ Transfering ∧ n >0 ⇒ n−1 ∈ Transfering
inv9 : ∀ n· n ∈ Transfering ⇒ 0 · · n ⊆ Transfering
inv10 : Transfering = dom(transfered)
inv11 : Transfering ≠ ∅ ⇒ Start_Transfering= TRUE
inv12 : last ∈ Transfering ⇔Transfered = TRUE
inv13 : last>0 ∧ last−1 ∈ Transfering ∧ last ∉ Transfering⇒ transfered ∪ {last ↦ FIN} = Send_Message
inv14 : last=0 ∧ last ∉ Transfering⇒ transfered ∪ {last ↦ FIN} = Send_Message
inv15 : last ∈ Transfering ⇒ transfered0 = transfered
inv16 : 0 ∉ Transfering ⇒ Transfering = ∅
inv17 : ∀ x,y · x ∈ ℕ ⇸ NET_MESSAGE ∧ y ⊆ x ∧ dom(x) = dom(y) ⇒ x = y
inv18 : ∀ n· n ∈ Transfering ∧ n+1 ∉ Transfering ⇒ 0 · · n = Transfering
EVENTS
INITIALISATION ≙
STATUS
ordinary
BEGIN
act1 : transfered ≔ ∅
act2 : Transfering ≔ ∅
act3 : Error ≔ FALSE
act4 : time ≔ 0
act5 : TransferingT ≔ ∅
act6 : Start_Transfering ≔ FALSE
act7 : Start_TransferingT ≔ 0
END

Start_Transfering ≙
extended
STATUS
ordinary
REFINES
Start_Transfering
WHEN
grd1 : *Start_Transfering = FALSE*
THEN
act1 : *Start_Transfering ≔TRUE*
act2 : *Start_TransferingT ≔time*
END

Transfering ≙
STATUS
ordinary
ANY

m
x
y
WHERE
grd1 : Start_Transfering = TRUE
grd2 : m ∈ Send_Message
grd3 : m = x ↦ y
grd4 : x ∉ Transfering
grd5 : x = 0 ⇒ Transfering= ∅
grd6 : x > 0 ⇒ x−1 ∈ Transfering
grd7 : Error = FALSE
grd8 : x < last
THEN
act1 : Transfering ≔ Transfering ∪ {x}
act2 : transfered ≔ transfered ∪ {m}
act3 : TransferingT(x) ≔ time
END

TransferingLast ≙
STATUS
ordinary
REFINES
Transfered
ANY
m
x
y
WHERE
grd1 : Start_Transfering = TRUE
grd2 : m ∈ Send_Message
grd3 : x = last
grd4 : m = x ↦ y
grd5 : x ∉ Transfering
grd6 : x > 0 ⇒ x−1 ∈ Transfering
grd7 : Error = FALSE
THEN
act1 : Transfering ≔ Transfering ∪ {x}
act2 : transfered ≔ transfered ∪ {m}
act3 : TransferingT(x) ≔ time
END

Error ≙
STATUS
ordinary
REFINES
Error
WHEN
grd1 : Start_Transfering = TRUE
grd2 : last ∉ Transfering
grd3 : Error = FALSE
THEN
act1 : Error ≔ TRUE
END

Tick_Tock ≙
STATUS
ordinary
REFINES
Tick_Tock
ANY
tick
WHERE
grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : Start_Transfering = TRUE ∧ last ∉ Transfering∧Error=FALSE⇒time+tick≤Start_TransferingT+DataTDL
THEN
act1 : time ≔ time+tick
END

END

Machine m02

MACHINE

m02

// *Deadline(Start_Transfering, Transfering(0)∨Error, PacketTDL)*

// *∀ x · x < last ⇒ Deadline(Transfering(x), Transfering(x+1)∨Error, PacketTDL)*

REFINES

m01

SEES

c01

VARIABLES

transfered // *Whole transfered Message*

Transfering // *Flag*

Error // *Flag*

time

TransferingT

Start_Transfering

Start_TransferingT

INVARIANTS

inv1 : Start_Transfering=TRUE ∧ 0 ∉ Transfering∧Error=FALSE ⇒time≤Start_TransferingT+ PacketTDL

inv2 : 0 ∈ Transfering ∧ Error=FALSE ⇒TransferingT(0)≤Start_TransferingT+ PacketTDL

inv3 : ∀ x · x < last ∧ x ∈ Transfering ∧ x+1 ∉ Transfering∧Error=FALSE ⇒time≤TransferingT(x)+ PacketTDL

inv4 : ∀ y · y ∈ Transfering ∧ Error = FALSE ⇒TransferingT(y)≤Start_TransferingT+(y+1)∗PacketTDL

inv5 : Start_TransferingT ≤ time

EVENTS

INITIALISATION ≙

extended

STATUS

ordinary

BEGIN

act1 : transfered ≔∅

act2 : Transfering ≔∅

act3 : Error ≔ FALSE

act4 : time ≔0

act5 : TransferingT ≔∅

act6 : Start_Transfering ≔FALSE

act7 : Start_TransferingT ≔0

END

Start_Transfering ≙

extended

STATUS

ordinary

REFINES

Start_Transfering

WHEN

grd1 : Start_Transfering = FALSE

THEN

act1 : Start_Transfering ≔TRUE

act2 : Start_TransferingT ≔time

END

Transfering ≙

STATUS

ordinary

REFINES

Transfering

TransferingLast

ANY

m

x

y

WHERE

grd1 : Start_Transfering = TRUE
grd2 : m ∈ Send_Message
grd3 : m = x ↦ y
grd4 : x ∉ Transfering
grd5 : x = 0 ⇒ Transfering= ∅
grd6 : x > 0 ⇒ x−1 ∈ Transfering
grd7 : Error = FALSE

THEN

act1 : Transfering ≔ Transfering ∪ {x}
act2 : transfered ≔ transfered ∪ {m}
act3 : TransferingT(x) ≔ time

END

Error ≙
extended
STATUS
ordinary

REFINES

Error

WHEN

grd1 : Start_Transfering = TRUE
grd2 : last ∉ Transfering
grd3 : Error = FALSE

THEN

act1 : Error ≔ TRUE

END

Tick_Tock ≙
STATUS
ordinary

REFINES

Tick_Tock

ANY

tick

WHERE

grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : Start_Transfering=TRUE∧0 ∉ Transfering∧Error=FALSE⇒time+tick≤Start_TransferingT+ PacketTDL
grd4 : ∀ x · x < last ∧ x ∈ Transfering ∧ x+1 ∉ Transfering∧Error=FALSE⇒time+tick≤TransferingT(x)+ PacketTDL

THEN

act1 : time ≔ time+tick

END

END

Machine m03

MACHINE

// *Deadline(Start_Transfering, Sending(0), 0)*

m03 // *∀ x · x<last ⇒ Deadline(Sending(x),Transfered(x) ∨ Error, PacketTDL)*

// *∀x · x =<last ⇒ Deadline(Transfered(x), Sending(x+1) ∨Error, 0)*

REFINES

m02

SEES

c01

VARIABLES

Transfering // *Flag*
TransferingT // *Flag*
Sending // *Flag*
Error // *Flag*
transfered
sent2 // *sent packets*
time // *time*
SendingT // *time of sending*

INVARIANTS

inv1 : Sending ⊆ dom(Send_Message) // *Type*
inv2 : finite(Sending)
inv3 : sent2 ⊆ Send_Message // *Type*
inv4 : SendingT ∈ Sending ⟶ ℕ // *Type*
inv5 : 0 ∈ Sending ⇔Start_Transfering = TRUE // *gluing invariant*
inv6 : 0 ∈ Sending ⇒ SendingT(0) = Start_TransferingT // *gluing invariant*
inv7 : ∀ n· n ∈ Sending ∧ n >0 ⇒ n−1 ∈ Transfering
inv8 : Transfering ⊆ Sending
inv9 : ∀ n· n ∈ Sending ∧ n >0 ⇒ n−1 ∈ Sending
inv10 : ∀ n· n ∈ Sending ⇒ 0 · · n ⊆ Sending
inv11 : ∀ n· n ∈ Sending ∧ n+1 ∉ Sending ⇒ 0 · · n = Sending
inv13 : ∀ x · x >0 ∧ x ∈ Sending ⇒ TransferingT(x−1) = SendingT(x)
inv14 : ∀ x · x ∈ Transfering ⇒ x ∈ Sending
inv15 : ∀ x · x <last ∧ x ∈ Transfering ∧ x+1 ∉ Sending ∧ Error = FALSE
⇒time≤TransferingT(x)+0
inv16 : ∀ x · x ∈ Transfering ⇒ TransferingT(x) ≤ time
inv17 : dom(sent2)=Sending
inv18 : 0 ∉ Sending ⇒ Sending = ∅
inv19 : ∀ x · x ∈dom(TransferingT) ⇒ time ≥ TransferingT(x)

EVENTS

INITIALISATION ≙
STATUS
ordinary

BEGIN

act1 : Sending ≔ ∅
act2 : Error ≔FALSE
act3 : transfered ≔∅
act4 : sent2 ≔∅
act5 : time ≔0
act6 : SendingT ≔∅
act7 : Transfering ≔∅
act8 : TransferingT ≔∅

END

Start_Transfering ≙
STATUS
ordinary

REFINES

Start_Transfering

ANY

m
x
y

WHERE

grd1 : m ∈ Send_Message

grd2 : m = x ↦ y
grd3 : x ∉ Sending
grd4 : x=0
grd5 : Error = FALSE
THEN
act1 : Sending ≔ Sending ∪ {x}
act2 : sent2 ≔ sent2 ∪ {m}
act3 : SendingT(x) ≔ time
END

Sending ≙
STATUS
ordinary
ANY
m
x
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x ∉ Sending
grd4 : x = 0 ⇒ Transfering= ∅
grd5 : x > 0 ∧ x−1 ∈ Transfering
grd6 : Error = FALSE
THEN
act1 : Sending ≔ Sending ∪ {x}
act2 : sent2 ≔ sent2 ∪ {m}
act3 : SendingT(x) ≔ time
END

Transfering ≙
STATUS
ordinary
REFINES
Transfering
ANY
m
x
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x ∉ Transfering
grd4 : x ∈ Sending
grd5 : Error = FALSE
THEN
act1 : Transfering ≔ Transfering ∪ {x}
act2 : transfered ≔ transfered ∪ {m}
act3 : TransferingT(x) ≔ time
END

Error ≙
STATUS
ordinary
REFINES
Error
ANY
x
WHERE
grd1 : x ∈ Sending
grd2 : x ∉ Transfering
grd3 : Error = FALSE
THEN
act1 : Error ≔ TRUE
END

Tick_Tock ≙
STATUS
ordinary

Message Passing Case-study (Time Added Manually)

REFINES
Tick_Tock
ANY
tick
WHERE
grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : ∀ x · x ∈ Sending ∧ x ∉ Transfering ∧ Error = FALSE
⇒time+tick≤SendingT(x)+PacketTDL
grd4 : ∀ x · x <last ∧ x ∈ Transfering ∧ x+1 ∉ Sending ∧ Error = FALSE
⇒time+tick≤TransferingT(x)+0
THEN
act1 : time ≔time+tick
END

END

Machine m04

MACHINE

m04

// *Deadline(Start_Transfering, Sending03(0↦0), 0)*

// *∀ x · x<= last ⇒ Deadline(Sending03(x↦0), Transfering03(x↦0)∨ResendingProcess(x↦1),ResendingDL)*

// *∀ x0,x1 · x1 < MaxResend ⇒ Deadline(ResendingProcess(x0↦x1), Transfering03(x0↦x1)∨ ResendingProcess(x0↦x1+1),ResendingDL)*

// *∀ x0 · x0<=last ⇒ Deadline(ResendingProcess(x0↦MaxResend) ,Transfering03(x0↦MaxResend)∨Error,ResendingDL)*

// *∀ x0,x1 · x0 <last ⇒ Deadline(Transfering03(x0↦x1) ,Sending03(x0+1↦0)∨ Error,0)*

// *∀ x0,x1 · x1 = 0 ⇒ Delay(Sending03(x0↦x1) ,ResendingProcess(x0↦x1+1),ResendingDL)*

// *∀ x0,x1 · x1 > 0 ∧ x1 < MaxResend ⇒ Delay(ResendingProcess(x0↦x1) ,ResendingProcess(x0↦x1+1),ResendingDL)*

// *∀ x0,x1 · x1 = MaxResend ⇒ Delay(ResendingProcess(x0↦x1) ,Error,ResendingDL)*

REFINES

m03

SEES

c02

VARIABLES

Sending03 // *Flag*

Error // *Flag*

Transfering03 // *Flag*

transfered

sent2 // *sent packets*

time

Sending03T

Transfering03T // *time*

ResendingProcess // *Flag*

ResendingProcessT // *time*

INVARIANTS

inv1 : ResendingProcess ∈ Sending ↔ 1‥MaxResend // *Type*

inv2 : ResendingProcessT ∈ ResendingProcess → ℕ // *Type*

inv3 : Sending03 ∈ Sending → {0} // *Type*

inv4 : Sending03T ∈ Sending03 → ℕ // *Type*

inv5 : ∀ x · x ∈ dom(SendingT) ⇒ Sending03T(x ↦ 0) = SendingT(x) // *gluing*

inv6 : Transfering03 ∈ Transfering → 0‥MaxResend // *Type*

inv7 : Transfering03T ∈ Transfering03 → ℕ // *Type*

inv8 : ∀ x,y · x ∈ dom(TransferingT) ∧ x↦ y ∈ dom(Transfering03T) ⇒ Transfering03T(x↦y) = TransferingT(x) // *gluing*

inv9 : ∀ x,y · x↦y ∈ ResendingProcess ∧ y > 1 ⇒ x↦y−1 ∈ ResendingProcess

inv10 : ∀ x · x↦1 ∈ ResendingProcess ⇒ x↦0 ∈ Sending03

inv11 : ∀ x0,x1 · x1 < MaxResend ∧ x0↦x1 ∈ ResendingProcess ∧ x0↦x1 ∉ Transfering03 ∧ x0↦x1+1 ∉ ResendingProcess
⇒time≤ResendingProcessT(x0↦x1)+ResendingDL

inv12 : ∀ x0 · x0↦MaxResend ∈ ResendingProcess ∧ x0↦MaxResend ∉ Transfering03 ∧ Error = FALSE
⇒time≤ResendingProcessT(x0↦MaxResend)+ResendingDL

inv13 : ∀ x · x↦0 ∈ Sending03 ∧ x↦0 ∉ Transfering03 ∧ x↦1 ∉ ResendingProcess
⇒time≤Sending03T(x↦0)+ResendingDL

inv14 : ∀ x0,x1 · x0↦x1 ∈ ResendingProcess ⇒ 1‥x1 ⊆ ran({x0}◁ResendingProcess)

inv15 : ∀ x0,x1 · x0↦x1 ∈ ResendingProcess⇒ResendingProcessT(x0↦x1)≤Sending03T(x0↦0)+ (x1) ∗ ResendingDL

inv16 : dom(ResendingProcess) ⊆ dom(Sending03)

EVENTS

INITIALISATION ≙

STATUS

ordinary

BEGIN

act1 : Sending03 ≔ ∅

act2 : Error ≔FALSE

act3 : Transfering03 ≔ ∅

act4 : transfered ≔∅

act5 : sent2 ≔∅

act6 : time ≔0

act7 : Sending03T ≔ ∅
act8 : Transfering03T ≔ ∅
act9 : ResendingProcess ≔ ∅
act10 : ResendingProcessT ≔ ∅
END

Sending03 ≙
STATUS
ordinary
REFINES
Sending
Start_Transfering
ANY
m
x
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x↦0 ∉ Sending03
grd4 : x = 0 ⇒Transfering03=∅
grd5 : x > 0 ⇒ x−1 ∈ dom(Transfering03)
grd6 : Error = FALSE
THEN
act1 : Sending03 ≔ Sending03 ∪ {x↦0}
act2 : sent2 ≔ sent2 ∪ {m}
act3 : Sending03T(x↦0) ≔ time
END

ResendingProcess ≙
STATUS
ordinary
ANY
x0
x1
WHERE
grd1 : x0↦x1−1 ∈ Sending03 ∪ ResendingProcess
grd2 : x0 ∉ dom(Transfering03)
grd3 : x0↦x1 ∉ ResendingProcess
grd4 : x1 ≤ MaxResend
grd5 : Error = FALSE
grd6 : x0↦x1−1 ∈ Sending03 ⇒ time ≥ Sending03T(x0↦x1−1) + ResendingDL
grd7 : x0↦x1−1 ∈ ResendingProcess ⇒ time ≥ ResendingProcessT(x0↦x1−1) + ResendingDL
THEN
act1 : ResendingProcess ≔ ResendingProcess ∪ {x0 ↦ x1}
act2 : ResendingProcessT(x0 ↦ x1) ≔ time
END

Transfering03 ≙
STATUS
ordinary
REFINES
Transfering
ANY
m
x0
x1
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd3 : x0 ∉ dom(Transfering03)
grd4 : x0 ↦ x1 ∈ Sending03 ∪ ResendingProcess
grd5 : Error = FALSE
WITH
x : x=x0
THEN
act1 : Transfering03 ≔ Transfering03 ∪ {x0 ↦ x1}
act2 : transfered ≔ transfered ∪ {m}

act3 : Transfering03T(x0 ↦ x1) ≔ time

END

Error ≙

STATUS

ordinary

REFINES

Error

ANY

x

y

WHERE

grd1 : x ↦ y ∈ ResendingProcess

grd2 : x ∉ dom(Transfering03)

grd3 : y = MaxResend

grd4 : Error = FALSE

grd5 : time ≥ ResendingProcessT(x↦y) + ResendingDL

THEN

act1 : Error ≔ TRUE

END

Tick_Tock ≙

STATUS

ordinary

REFINES

Tick_Tock

ANY

tick

WHERE

grd1 : tick∈ℕ

grd2 : tick > 0

grd3 : ∀ x · x↦0 ∈ Sending03 ∧ x↦0 ∉ Transfering03 ∧ x↦1 ∉ ResendingProcess
⇒time+tick≤Sending03T(x↦0)+ResendingDL

grd4 : ∀ x0,x1 · x1 < MaxResend ∧ x0↦x1 ∈ ResendingProcess ∧ x0↦x1 ∉ Transfering03 ∧ x0↦x1+1 ∉ Resending Process
⇒time+tick≤ResendingProcessT(x0↦x1)+ResendingDL

grd5 : ∀ x0 · x0↦MaxResend ∈ ResendingProcess ∧ x0↦MaxResend ∉ Transfering03 ∧ Error = FALSE
⇒time+tick≤ResendingProcessT(x0↦MaxResend)+ResendingDL

grd6 : ∀ x0,x1 · x0 <last ∧ x0↦x1 ∈ Transfering03 ∧ x0+1 ↦0 ∉ Sending03 ∧ Error = FALSE
⇒time+tick≤Transfering03T(x0↦x1)+0

THEN

act1 : time ≔ time+tick

END

END

Machine m05

MACHINE

// @grd3 Deadline(Start_Transfering, {Sending03(0↦0)}, 0)

// @grd4 ∀ x · x <= last ∧ Deadline(Sending03(x↦0), {Transfered03(x↦0),ResendingProcess(x↦1)},ResendingDL)

m05 // @grd5 ∀ x0,x1 · x1 < MaxResend ∧ Deadline(ResendingProcess(x0↦x1), {Transfered03(x0↦x1), ResendingProcess(x0↦x1+1)},ResendingDL)

// @grd6 ∀ x0 ·x0 <=last ∧ Deadline(ResendingProcess(x0↦MaxResend) ,{Transfered03(x0↦MaxResend),Error},ResendingDL)

// @grd7 ∀ x0,x1 · x0 <last ∧ Deadline(Transfered03(x0↦x1) ,{Sending03(x0+1↦0), Error},0)

REFINES

m04

SEES

c02

VARIABLES

Sending03 // Flag
Error // Flag
Transfering03 // Flag
transfered
sent2 // sent packets
time
Sending03T
Transfering03T // time
ResendingProcess // Flag
ResendingProcessT // time
DataLost // Flag
ACKLost // Flag

INVARIANTS

inv1 : DataLost ⊆ Sending03 ∪ ResendingProcess
inv2 : ACKLost ⊆ Sending03 ∪ ResendingProcess

EVENTS

INITIALISATION ≙
extended
STATUS
ordinary

BEGIN

act1 : Sending03 ≔ ∅
act2 : Error ≔FALSE
act3 : Transfering03 ≔ ∅
act4 : transfered ≔∅
act5 : sent2 ≔∅
act6 : time ≔0
act7 : Sending03T ≔∅
act8 : Transfering03T ≔∅
act9 : ResendingProcess ≔∅
act10 : ResendingProcessT ≔∅
act13 : DataLost ≔∅
act14 : ACKLost ≔∅

END

Sending03 ≙
extended
STATUS
ordinary

REFINES

Sending03

ANY

m
x
y

WHERE

grd1 : m ∈ Send_Message
grd2 : m = x ↦y
grd3 : x↦0 ∉ Sending03

grd4 : $x = 0 \Rightarrow Transfering03 = \emptyset$
grd5 : $x > 0 \Rightarrow x-1 \in dom(Transfering03)$
grd6 : $Error = FALSE$
THEN
act1 : $Sending03 := Sending03 \cup \{x \mapsto 0\}$
act2 : $sent2 := sent2 \cup \{m\}$
act3 : $Sending03T(x \mapsto 0) := time$
END

DataLost ≙
STATUS
ordinary
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1 \in Sending03 \cup ResendingProcess$
grd2 : $x0 \mapsto x1 \notin Transfering03$
grd3 : $x0 \mapsto x1 \notin DataLost$
grd4 : $x0 \mapsto x1 \notin ACKLost$
THEN
act1 : $DataLost := DataLost \cup \{x0 \mapsto x1\}$
END

ACKLost ≙
STATUS
ordinary
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1 \in Sending03 \cup ResendingProcess$
grd2 : $x0 \mapsto x1 \notin Transfering03$
grd3 : $x0 \mapsto x1 \notin DataLost$
grd4 : $x0 \mapsto x1 \notin ACKLost$
THEN
act1 : $ACKLost := ACKLost \cup \{x0 \mapsto x1\}$
END

ResendingProcess ≙
extended
STATUS
ordinary
REFINES
ResendingProcess
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1-1 \in Sending03 \cup ResendingProcess$
grd2 : $x0 \notin dom(Transfering03)$
grd3 : $x0 \mapsto x1 \notin ResendingProcess$
grd4 : $x1 \leq MaxResend$
grd5 : $Error = FALSE$
grd6 : $x0 \mapsto x1-1 \in Sending03 \Rightarrow time \geq Sending03T(x0 \mapsto x1-1) + ResendingDL$
grd7 : $x0 \mapsto x1-1 \in ResendingProcess \Rightarrow time \geq ResendingProcessT(x0 \mapsto x1-1) + ResendingDL$
THEN
act1 : $ResendingProcess := ResendingProcess \cup \{x0 \mapsto x1\}$
act2 : $ResendingProcessT(x0 \mapsto x1) := time$
END

Error ≙
extended
STATUS
ordinary
REFINES
Error
ANY

x
y
WHERE
grd1 : $x \mapsto y \in ResendingProcess$
grd2 : $x \notin dom(Transfering03)$
grd3 : $y = MaxResend$
grd4 : $Error = FALSE$
grd5 : $time \geq ResendingProcessT(x \mapsto y) + ResendingDL$
THEN
act1 : $Error := TRUE$
END

Tick_Tock ≙
extended
STATUS
ordinary
REFINES
Tick_Tock
ANY
tick
WHERE
grd1 : $tick \in \mathbb{N}$
grd2 : $tick > 0$
grd3 : $\forall x \cdot x \mapsto 0 \in Sending03 \land x \mapsto 0 \notin Transfering03 \land x \mapsto 1 \notin ResendingProcess \Rightarrow time+tick \leq Sending03T(x \mapsto 0)+ResendingDL$
grd4 : $\forall x0,x1 \cdot x1 < MaxResend \land x0 \mapsto x1 \in ResendingProcess \land x0 \mapsto x1 \notin Transfering03 \land x0 \mapsto x1+1 \notin Resending Process \Rightarrow time+tick \leq ResendingProcessT(x0 \mapsto x1)+ResendingDL$
grd5 : $\forall x0 \cdot x0 \mapsto MaxResend \in ResendingProcess \land x0 \mapsto MaxResend \notin Transfering03 \land Error = FALSE \Rightarrow time+tick \leq ResendingProcessT(x0 \mapsto MaxResend)+ResendingDL$
grd6 : $\forall x0,x1 \cdot x0 < last \land x0 \mapsto x1 \in Transfering03 \land x0+1 \mapsto 0 \notin Sending03 \land Error = FALSE \Rightarrow time+tick \leq Transfering03T(x0 \mapsto x1)+0$
THEN
act1 : $time := time+tick$
END

Transfering03 ≙
extended
STATUS
ordinary
REFINES
Transfering03
ANY
m
x0
x1
y
WHERE
grd1 : $m \in Send\_Message$
grd2 : $m = x0 \mapsto y$
grd3 : $x0 \notin dom(Transfering03)$
grd4 : $x0 \mapsto x1 \in Sending03 \cup ResendingProcess$
grd5 : $Error = FALSE$
THEN
act1 : $Transfering03 := Transfering03 \cup \{x0 \mapsto x1\}$
act2 : $transfered := transfered \cup \{m\}$
act3 : $Transfering03T(x0 \mapsto x1) := time$
END

END

Machine m06

MACHINE

m06

// *Deadline(Start_Transfering, Sending03(0↦0), 0)*

// *∀ x · x ≤ last ⇒ Deadline(Sending03(x↦0), Transfered03(x↦0)∨ResendingProcess(x↦1),ResendingDL)*

// *∀ x0,x1 · x1 < MaxResend ⇒ Deadline(ResendingProcess(x0↦x1), Transfered03(x0↦x1)∨ ResendingProcess(x0↦x1+1),ResendingDL)*

// *∀ x0 · x ≤ last ⇒ Deadline(ResendingProcess(x0↦MaxResend) ,Transfered03(x0↦MaxResend)∨Error,ResendingDL)*

// *∀ x0,x1 · x0 <last ⇒ Deadline(Transfered03(x0↦x1) ,Sending03(x0+1↦0)∨Error,0)*

// *∀ x · x ≤ last ∧ y ≤ MaxResend ⇒ Deadline(Sending03(x↦0), Receiving(x↦0)∨DataLost(x↦0), ChannelDL)*

// *∀ x,y · x ≤ last ∧ y ≤ MaxResend ⇒ Deadline(Resending(x↦y), Receiving(x↦y)∨Rereceiving(x↦y)∨DataLost(x↦y),ChannelDL)*

// *∀ x,y · x ≤ last ∧ y ≤ MaxResend ⇒ Deadline(Receiving(x↦y), ReceivingACK(x↦y)∨ACKLost(x↦y),ChannelDL)*

// *∀ x,y · x ≤ last ∧ y ≤ MaxResend ⇒ Deadline(Rereceiving(x↦y),ReceivingACK(x↦y)∨ACKLost(x↦y), ChannelDL)*

REFINES

m05

SEES

c03

VARIABLES

Sending03 // *Flag*
Error // *Flag*
ReceivingACK // *Flag*
transfered
sent2 // *sent packets*
time
Sending03T
ReceivingACKT // *time*
Resending // *Flag*
ResendingT // *time*
DataLost // *Flag*
ACKLost // *Flag*
Receiving // *Flag*
ReceivingT // *time*
received // *Keep track of the received packetes in the receiver*
Rereceiving // *Flag*
RereceivingT // *time*
Receiver_Error // *Flag*

INVARIANTS

inv1 : Transfering03=ReceivingACK // *Gluing*
inv2 : Transfering03T= ReceivingACKT // *Gluing*
inv3 : Receiving ∈ dom(Sending03) ⇸ 0‥MaxResend // *Type*
inv4 : ReceivingT ∈ Receiving ⟶ ℕ // *Type*
inv5 : received ⊆ sent2 // *Type*
inv6 : ResendingProcess = Resending
inv7 : ResendingProcessT = ResendingT
inv8 : Rereceiving ⊆ Resending // *Type*
inv9 : RereceivingT ∈ Rereceiving ⟶ ℕ // *Type*
inv10 : Receiving ⊆ Sending03 ∪ Resending
inv11 : ACKLost ⊆ Receiving ∪ Rereceiving
inv12 : ∀ x · x ∈ Receiving ∪ Rereceiving ⇒ x ∉ DataLost
inv13 : ReceivingACK ⊆ Receiving ∪ Rereceiving
inv14 : Receiver_Error ∈ BOOL // *Type*
inv15 : ∀ x · x↦0 ∈ Sending03 ∧ x↦1 ∈ Resending ⇒ ResendingT(x↦1) ≥ Sending03T(x↦0) + ResendingDL
inv16 : ∀ x,y · x↦y−1 ∈ Resending ∧ x↦y ∈ Resending ⇒ ResendingT(x↦y) ≥ ResendingT(x↦y−1) + ResendingDL

EVENTS

INITIALISATION ≙
STATUS
ordinary

BEGIN

act1 : Sending03 ≔ ∅

act2 : Error ≔ FALSE
act3 : ReceivingACK ≔ ∅
act4 : transfered ≔ ∅
act5 : sent2 ≔ ∅
act6 : time ≔ 0
act7 : Sending03T ≔ ∅
act8 : ReceivingACKT ≔ ∅
act9 : Resending ≔ ∅
act10 : ResendingT ≔ ∅
act11 : DataLost ≔ ∅
act12 : ACKLost ≔ ∅
act13 : Receiving ≔ ∅
act14 : ReceivingT ≔ ∅
act15 : received ≔ ∅
act16 : Rereceiving ≔ ∅
act17 : RereceivingT ≔ ∅
act18 : Receiver_Error ≔ FALSE

END

Sending03 ≙
STATUS
ordinary
REFINES
Sending03
ANY
m
x
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x↦0 ∉ Sending03
grd4 : x = 0 ⇒ReceivingACK=∅
grd5 : x > 0 ⇒ x−1 ∈ dom(ReceivingACK)
grd6 : Error = FALSE
THEN
act1 : Sending03 ≔ Sending03 ∪ {x↦0}
act2 : sent2 ≔ sent2 ∪ {m}
act3 : Sending03T(x↦0) ≔ time
END

DataLost ≙
STATUS
ordinary
REFINES
DataLost
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Sending03 ∪ Resending
grd2 : x0↦x1 ∉ Receiving ∪ Rereceiving
grd3 : x0↦x1 ∉ DataLost
THEN
act1 : DataLost ≔ DataLost ∪ {x0 ↦ x1}
END

Receiving ≙
STATUS
ordinary
ANY
m
x0
x1
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y

grd3 : x0↦x1 ∈ Resending ∪ Sending03
grd4 : x0 ∉ dom(Receiving)
grd5 : x0↦x1 ∉ DataLost
THEN
act1 : Receiving ≔ Receiving ∪ {x0↦x1}
act2 : received ≔ received ∪ {m}
act3 : ReceivingT(x0↦x1) ≔ time
END

Rereceiving ≙
STATUS
ordinary
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Resending
grd2 : x0↦x1 ∉ DataLost
grd3 : x0↦x1 ∉ Rereceiving
grd4 : x0↦x1 ∉ Receiving
grd5 : x0 ∈ dom(Receiving)
THEN
act1 : Rereceiving ≔ Rereceiving ∪ {x0↦x1}
act2 : RereceivingT(x0↦x1) ≔ time
END

ACKLost ≙
STATUS
ordinary
REFINES
ACKLost
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Receiving ∪ Rereceiving
grd2 : x0↦x1 ∉ ReceivingACK
grd3 : x0↦x1 ∉ ACKLost
THEN
act1 : ACKLost ≔ ACKLost ∪ {x0 ↦ x1}
END

Resending ≙
STATUS
ordinary
REFINES
ResendingProcess
ANY
x0
x1
WHERE
grd1 : x0↦x1−1 ∈ Sending03 ∪ Resending
grd2 : x0 ∉ dom(ReceivingACK)
grd3 : x0↦x1 ∉ Resending
grd4 : x1 ≤ MaxResend
grd5 : Error = FALSE
grd6 : x0↦x1−1 ∈ Sending03 ⇒ time ≥ Sending03T(x0↦x1−1) + ResendingDL
grd7 : x0↦x1−1 ∈ Resending ⇒ time ≥ ResendingT(x0↦x1−1) + ResendingDL
THEN
act1 : Resending ≔ Resending ∪ {x0 ↦ x1}
act2 : ResendingT(x0 ↦ x1) ≔ time
END

ReceivingACK ≙
STATUS
ordinary
REFINES
Transfering03

ANY
m
x0
x1
y
WHERE
grd1 : $m \in Send\_Message$
grd2 : $m = x0 \mapsto y$
grd3 : $x0 \notin dom(ReceivingACK)$
grd4 : $x0 \mapsto x1 \in Receiving \cup Rereceiving$
grd5 : $x0 \mapsto x1 \notin ACKLost$
grd6 : $Error = FALSE$
THEN
act1 : $ReceivingACK \coloneqq ReceivingACK \cup \{x0 \mapsto x1\}$
act2 : $transfered \coloneqq transfered \cup \{m\}$
act3 : $ReceivingACKT(x0 \mapsto x1) \coloneqq time$
END

Error ≙
STATUS
ordinary
REFINES
Error
ANY
x
y
WHERE
grd1 : $x \mapsto y \in Resending$
grd2 : $x \notin dom(ReceivingACK)$
grd3 : $y = MaxResend$
grd4 : $Error = FALSE$
grd5 : $time \geq ResendingT(x \mapsto y) + ResendingDL$
THEN
act1 : $Error \coloneqq TRUE$
END

Receiver_Error ≙
STATUS
ordinary
ANY
x
y
WHERE
grd1 : $Receiver\_Error = FALSE$
grd2 : $Receiving \neq \emptyset$
grd3 : $last \notin dom(Receiving)$
grd4 : $x \mapsto y \in Receiving$
grd5 : $x < last$
grd6 : $x+1 \mapsto y \notin Receiving$
grd7 : $time \geq ReceivingT(x \mapsto y) + MaxResend * ResendingDL$
THEN
act1 : $Receiver\_Error \coloneqq TRUE$
END

Tick_Tock ≙
STATUS
ordinary
REFINES
Tick_Tock
ANY
tick
WHERE
grd1 : $tick \in \mathbb{N}$
grd2 : $tick > 0$
grd3 : $\forall x \cdot x \mapsto 0 \in Sending03 \wedge x \mapsto 0 \notin ReceivingACK \wedge x \mapsto 1 \notin Resending \Rightarrow time+tick \leq Sending03T(x \mapsto 0)+ResendingDL$
grd4 : $\forall x0,x1 \cdot x1 < MaxResend \wedge x0 \mapsto x1 \in Resending \wedge x0 \mapsto x1 \notin ReceivingACK \wedge x0 \mapsto x1+1 \notin Resending \Rightarrow time+tick \leq ResendingT(x0 \mapsto x1)+ResendingDL$

grd5 : ∀ x0 · x0↦MaxResend ∈ Resending ∧ x0↦MaxResend ∉ ReceivingACK ∧ Error = FALSE
⇒time+tick≤ResendingT(x0↦MaxResend)+ResendingDL

grd6 : ∀ x0,x1 · x0 <last ∧ x0↦x1 ∈ ReceivingACK ∧ x0+1 ↦0 ∉ Sending03 ∧ Error = FALSE
⇒time+tick≤ReceivingACKT(x0↦x1)+0

grd7 : ∀ x · x↦0 ∈ Sending03 ∧ x↦0 ∉ Receiving ∧ x↦0 ∉ DataLost ⇒ time+tick ≤ Sending03T(x↦0) + ChannelDL // Channel

grd8 : ∀ x,y · x↦y ∈ Resending ∧ x↦y ∉ Receiving ∧ x↦y ∉ Rereceiving ∧ x↦y ∉ DataLost ⇒ time+tick ≤ ResendingT(x↦y) + ChannelDL // Channel

grd9 : ∀ x,y · x↦y ∈ Receiving ∧ x↦y ∉ ReceivingACK ∧ x↦y ∉ ACKLost ⇒ time+tick ≤ ReceivingT(x↦y) + ChannelDL // Channel

grd10 : ∀ x,y · x↦y ∈ Rereceiving ∧ x↦y ∉ ReceivingACK ∧ x↦y ∉ ACKLost ⇒ time+tick ≤ RereceivingT(x↦y) + ChannelDL // Channel

grd11 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Receiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time+tick≤ReceivingT(x0↦x1)+ReceiverDL // Receiver

grd12 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Rereceiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time+tick≤RereceivingT(x0↦x1)+ReceiverDL // Receiver

THEN

act1 : time ≔ time+tick

END

END

Machine m07

MACHINE

m07 // *Added the created order between drop and resend based on delay*

REFINES

m06

SEES

c03

VARIABLES

Sending03 // *Flag*
Error // *Flag*
ReceivingACK // *Flag*
transfered
sent2 // *sent packets*
time
Sending03T
ReceivingACKT // *time*
Resending // *Flag*
ResendingT // *time*
DataLost // *Flag*
ACKLost // *Flag*
Receiving // *Flag*
ReceivingT // *time*
received // *Keep track of the received packetes in the receiver*
Rereceiving // *Flag*
RereceivingT // *time*
Receiver_Error // *Flag*

INVARIANTS

inv1 : $\forall x \cdot x \mapsto 0 \in Sending03 \land x \mapsto 0 \notin Receiving \land x \mapsto 0 \notin DataLost \Rightarrow time \leq Sending03T(x \mapsto 0) + ChannelDL$ // *Channel*

inv2 : $\forall x,y \cdot x \mapsto y \in Resending \land x \mapsto y \notin Receiving \land x \mapsto y \notin Rereceiving \land x \mapsto y \notin DataLost \Rightarrow time \leq ResendingT(x \mapsto y) + ChannelDL$ // *Channel*

inv3 : $\forall x,y \cdot x \mapsto y \in Receiving \land x \mapsto y \notin ReceivingACK \land x \mapsto y \notin ACKLost \Rightarrow time \leq ReceivingT(x \mapsto y) + ChannelDL$ // *Channel*

inv4 : $\forall x,y \cdot x \mapsto y \in Rereceiving \land x \mapsto y \notin ReceivingACK \land x \mapsto y \notin ACKLost \Rightarrow time \leq RereceivingT(x \mapsto y) + ChannelDL$ // *Channel*

inv5 : $\forall x \cdot x \mapsto 0 \in Sending03 \land x \mapsto 0 \in Receiving \Rightarrow ReceivingT(x \mapsto 0) \leq Sending03T(x \mapsto 0) + ChannelDL$

inv6 : $\forall x,y \cdot x \mapsto y \in Resending \land x \mapsto y \in Receiving \Rightarrow ReceivingT(x \mapsto y) \leq ResendingT(x \mapsto y) + ChannelDL$

inv7 : $\forall x,y \cdot x \mapsto y \in Rereceiving \Rightarrow x \in dom(Receiving)$

inv8 : $\forall x,y \cdot x \mapsto y \in Resending \land x \mapsto y \in Rereceiving \Rightarrow RereceivingT(x \mapsto y) \leq ResendingT(x \mapsto y) + ChannelDL$

inv9 : $\forall x,y \cdot x \mapsto y \in Rereceiving \Rightarrow x \mapsto y \in Resending$

inv17 : $\forall x,y \cdot x \mapsto y \in Resending \Rightarrow (x \mapsto y-1 \in Receiving \cup Rereceiving \land x \mapsto y-1 \in ACKLost) \lor (x \mapsto y-1 \in DataLost)$

EVENTS

INITIALISATION $\triangleq$

extended

STATUS

ordinary

BEGIN

act1 : *Sending03* $\coloneqq \emptyset$
act2 : *Error* $\coloneqq$ *FALSE*
act3 : *ReceivingACK* $\coloneqq \emptyset$
act4 : *transfered* $\coloneqq \emptyset$
act5 : *sent2* $\coloneqq \emptyset$
act6 : *time* $\coloneqq 0$
act7 : *Sending03T* $\coloneqq \emptyset$
act8 : *ReceivingACKT* $\coloneqq \emptyset$
act9 : *Resending* $\coloneqq \emptyset$
act10 : *ResendingT* $\coloneqq \emptyset$
act11 : *DataLost* $\coloneqq \emptyset$
act12 : *ACKLost* $\coloneqq \emptyset$
act13 : *Receiving* $\coloneqq \emptyset$
act14 : *ReceivingT* $\coloneqq \emptyset$
act15 : *received* $\coloneqq \emptyset$
act16 : *Rereceiving* $\coloneqq \emptyset$

act17 : RereceivingT ≔ ∅
act18 : Receiver_Error ≔ FALSE
END

Sending03 ≙
extended
STATUS
ordinary
REFINES
Sending03
ANY
m
x
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x↦0 ∉ Sending03
grd4 : x = 0 ⇒ReceivingACK= ∅
grd5 : x > 0 ⇒x−1 ∈ dom(ReceivingACK)
grd6 : Error = FALSE
THEN
act1 : Sending03 ≔Sending03 ∪ {x↦0}
act2 : sent2 ≔sent2 ∪ {m}
act3 : Sending03T(x↦0) ≔time
END

DataLost ≙
extended
STATUS
ordinary
REFINES
DataLost
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Sending03 ∪ Resending
grd2 : x0↦x1 ∉ Receiving ∪ Rereceiving
grd3 : x0↦x1 ∉ DataLost
THEN
act1 : DataLost ≔DataLost ∪ {x0 ↦x1}
END

Receiving ≙
extended
STATUS
ordinary
REFINES
Receiving
ANY
m
x0
x1
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦y
grd3 : x0↦x1 ∈ Resending ∪ Sending03
grd4 : x0 ∉ dom(Receiving)
grd5 : x0↦x1 ∉ DataLost
THEN
act1 : Receiving ≔Receiving ∪ {x0↦x1}
act2 : received ≔received ∪ {m}
act3 : ReceivingT(x0↦x1) ≔time
END

Rereceiving ≙
extended

STATUS
ordinary
REFINES
Rereceiving
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1 \in Resending$
grd2 : $x0 \mapsto x1 \notin DataLost$
grd3 : $x0 \mapsto x1 \notin Rereceiving$
grd4 : $x0 \mapsto x1 \notin Receiving$
grd5 : $x0 \in dom(Receiving)$
THEN
act1 : $Rereceiving := Rereceiving \cup \{x0 \mapsto x1\}$
act2 : $RereceivingT(x0 \mapsto x1) := time$
END

ACKLost ≙
extended
STATUS
ordinary
REFINES
ACKLost
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1 \in Receiving \cup Rereceiving$
grd2 : $x0 \mapsto x1 \notin ReceivingACK$
grd3 : $x0 \mapsto x1 \notin ACKLost$
THEN
act1 : $ACKLost := ACKLost \cup \{x0 \mapsto x1\}$
END

Resending ≙
extended
STATUS
ordinary
REFINES
Resending
ANY
x0
x1
WHERE
grd1 : $x0 \mapsto x1-1 \in Sending03 \cup Resending$
grd2 : $x0 \notin dom(ReceivingACK)$
grd3 : $x0 \mapsto x1 \notin Resending$
grd4 : $x1 \leq MaxResend$
grd5 : $Error = FALSE$
grd6 : $x0 \mapsto x1-1 \in Sending03 \Rightarrow time \geq Sending03T(x0 \mapsto x1-1) + ResendingDL$
grd7 : $x0 \mapsto x1-1 \in Resending \Rightarrow time \geq ResendingT(x0 \mapsto x1-1) + ResendingDL$
THEN
act1 : $Resending := Resending \cup \{x0 \mapsto x1\}$
act2 : $ResendingT(x0 \mapsto x1) := time$
END

ReceivingACK ≙
extended
STATUS
ordinary
REFINES
ReceivingACK
ANY
m
x0
x1
y
WHERE

grd1 : $m \in Send\_Message$
grd2 : $m = x0 \mapsto y$
grd3 : $x0 \notin dom(ReceivingACK)$
grd4 : $x0 \mapsto x1 \in Receiving \cup Rereceiving$
grd5 : $x0 \mapsto x1 \notin ACKLost$
grd6 : $Error = FALSE$

THEN

act1 : $ReceivingACK := ReceivingACK \cup \{x0 \mapsto x1\}$
act2 : $transfered := transfered \cup \{m\}$
act3 : $ReceivingACKT(x0 \mapsto x1) := time$

END

Error ≙

extended
STATUS
ordinary

REFINES
Error

ANY
x
y

WHERE

grd1 : $x \mapsto y \in Resending$
grd2 : $x \notin dom(ReceivingACK)$
grd3 : $y = MaxResend$
grd4 : $Error = FALSE$
grd5 : $time \geq ResendingT(x \mapsto y) + ResendingDL$

THEN

act1 : $Error := TRUE$

END

Receiver_Error ≙

extended
STATUS
ordinary

REFINES
Receiver_Error

ANY
x
y

WHERE

grd1 : $Receiver\_Error = FALSE$
grd2 : $Receiving \neq \emptyset$
grd3 : $last \notin dom(Receiving)$
grd4 : $x \mapsto y \in Receiving$
grd5 : $x < last$
grd6 : $x+1 \mapsto y \notin Receiving$
grd7 : $time \geq ReceivingT(x \mapsto y) + MaxResend * ResendingDL$

THEN

act1 : $Receiver\_Error := TRUE$

END

Tick_Tock ≙

extended
STATUS
ordinary

REFINES
Tick_Tock

ANY
$tick$

WHERE

grd1 : $tick \in \mathbb{N}$
grd2 : $tick > 0$
grd3 : $\forall x \cdot x \mapsto 0 \in Sending03 \wedge x \mapsto 0 \notin ReceivingACK \wedge x \mapsto 1 \notin Resending \Rightarrow time+tick \leq Sending03T(x \mapsto 0)+ResendingDL$
grd4 : $\forall x0,x1 \cdot x1 < MaxResend \wedge x0 \mapsto x1 \in Resending \wedge x0 \mapsto x1 \notin ReceivingACK \wedge x0 \mapsto x1+1 \notin Resending \Rightarrow time+tick \leq ResendingT(x0 \mapsto x1)+ResendingDL$
grd5 : $\forall x0 \cdot x0 \mapsto MaxResend \in Resending \wedge x0 \mapsto MaxResend \notin ReceivingACK \wedge Error = FALSE$

⇒time+tick≤ResendingT(x0↦MaxResend)+ResendingDL

grd6 : ∀x0,x1 · x0 <last ∧ x0↦x1 ∈ ReceivingACK ∧ x0+1 ↦0 ∉ Sending03 ∧ Error = FALSE
⇒time+tick≤ReceivingACKT(x0↦x1)+0

grd7 : ∀x · x↦0 ∈ Sending03 ∧ x↦0 ∉ Receiving ∧ x↦0 ∉ DataLost ⇒time+tick ≤ Sending03T(x↦0) + ChannelDL // Channel

grd8 : ∀x,y · x↦y ∈ Resending ∧ x↦y ∉ Receiving ∧ x↦y ∉ Rereceiving ∧ x↦y ∉ DataLost ⇒time+tick ≤ ResendingT(x↦y) + ChannelDL // Channel

grd9 : ∀x,y · x↦y ∈ Receiving ∧ x↦y ∉ ReceivingACK ∧ x↦y ∉ ACKLost ⇒time+tick ≤ ReceivingT(x↦y) + ChannelDL // Channel

grd10 : ∀x,y · x↦y ∈ Rereceiving ∧ x↦y ∉ ReceivingACK ∧ x↦y ∉ ACKLost ⇒time+tick ≤ RereceivingT(x↦y) + ChannelDL // Channel

grd11 : ∀x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Receiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time+tick≤ReceivingT(x0↦x1)+ReceiverDL // Receiver

grd12 : ∀x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Rereceiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time+tick≤RereceivingT(x0↦x1)+ReceiverDL // Receiver

THEN

act1 : time ≔time+tick

END

END

Machine m08

MACHINE
m08 // *Duplicate variables to decompose*
REFINES
m07
SEES
c03
VARIABLES
Sending03 // *Flag*
Error // *Flag*
ReceivingACK // *Flag*
transfered
sent2 // *sent packets*
time
Sending03T
ReceivingACKT // *time*
Resending // *Flag*
ResendingT // *time*
DataLost // *Flag*
ACKLost // *Flag*
Receiving // *Flag*
ReceivingT // *time*
received // *Keep track of the received packetes in the receiver*
Rereceiving // *Flag*
RereceivingT // *time*
Receiver_Error // *Type*
Sending03_Channel // *duplicate*
Receiving_Channel // *duplicate*
Sending03T_Channel // *duplicate*
Resending_Channel // *duplicate*
Rereceiving_Channel // *duplicate*
ReceivingACK_Channel // *duplicate*
ReceivingT_Channel // *duplicate*
RereceivingT_Channel // *duplicate*
ResendingT_Channel // *duplicate*
ReceivingACKT_Channel // *duplicate*
time_channel // *duplicate*
time_receiver // *duplicate*
INVARIANTS
inv1 : Sending03_Channel = Sending03
inv2 : Receiving_Channel = Receiving
inv3 : Sending03T_Channel = Sending03T
inv4 : Resending_Channel = Resending
inv5 : Rereceiving_Channel = Rereceiving
inv6 : ReceivingACK_Channel = ReceivingACK
inv7 : ReceivingT_Channel = ReceivingT
inv8 : RereceivingT_Channel = RereceivingT
inv9 : ResendingT_Channel = ResendingT
inv10 : ReceivingACKT_Channel = ReceivingACKT
inv11 : time_channel = time
inv12 : time_receiver = time
EVENTS
INITIALISATION ≙
extended
STATUS
ordinary
BEGIN
act1 : *Sending03* ≔ ∅
act2 : *Error* ≔*FALSE*
act3 : *ReceivingACK* ≔ ∅
act4 : *transfered* ≔∅
act5 : *sent2* ≔∅
act6 : *time* ≔*0*
act7 : *Sending03T* ≔∅
act8 : *ReceivingACKT* ≔∅
act9 : *Resending* ≔∅

act10 : $ResendingT := \emptyset$
act11 : $DataLost := \emptyset$
act12 : $ACKLost := \emptyset$
act13 : $Receiving := \emptyset$
act14 : $ReceivingT := \emptyset$
act15 : $received := \emptyset$
act16 : $Rereceiving := \emptyset$
act17 : $RereceivingT := \emptyset$
act18 : $Receiver\_Error := FALSE$
act19 : Sending03_Channel $:= \emptyset$
act20 : Receiving_Channel $:= \emptyset$
act21 : Sending03T_Channel $:= \emptyset$
act22 : Resending_Channel $:= \emptyset$
act23 : Rereceiving_Channel $:= \emptyset$
act24 : ReceivingACK_Channel $:= \emptyset$
act25 : ReceivingT_Channel $:= \emptyset$
act26 : RereceivingT_Channel $:= \emptyset$
act27 : ResendingT_Channel $:= \emptyset$
act28 : ReceivingACKT_Channel $:= \emptyset$
act29 : time_receiver $:= 0$
act30 : time_channel $:= 0$

END

Sending03 ≙
extended
STATUS
ordinary

REFINES
Sending03

ANY
m
x
y

WHERE
grd1 : $m \in Send\_Message$
grd2 : $m = x \mapsto y$
grd3 : $x \mapsto 0 \notin Sending03$
grd4 : $x = 0 \Rightarrow ReceivingACK = \emptyset$
grd5 : $x > 0 \Rightarrow x-1 \in dom(ReceivingACK)$
grd6 : $Error = FALSE$

THEN
act1 : $Sending03 := Sending03 \cup \{x \mapsto 0\}$
act2 : $sent2 := sent2 \cup \{m\}$
act3 : $Sending03T(x \mapsto 0) := time$
act4 : Sending03_Channel := Sending03_Channel ∪ {x↦0}
act5 : Sending03T_Channel(x↦0) := time_channel

END

Resending ≙
extended
STATUS
ordinary

REFINES
Resending

ANY
$x0$
$x1$

WHERE
grd1 : $x0 \mapsto x1-1 \in Sending03 \cup Resending$
grd2 : $x0 \notin dom(ReceivingACK)$
grd3 : $x0 \mapsto x1 \notin Resending$
grd4 : $x1 \leq MaxResend$
grd5 : $Error = FALSE$
grd6 : $x0 \mapsto x1-1 \in Sending03 \Rightarrow time \geq Sending03T(x0 \mapsto x1-1) + ResendingDL$
grd7 : $x0 \mapsto x1-1 \in Resending \Rightarrow time \geq ResendingT(x0 \mapsto x1-1) + ResendingDL$

THEN
act1 : $Resending := Resending \cup \{x0 \mapsto x1\}$
act2 : $ResendingT(x0 \mapsto x1) := time$

act3 : Resending_Channel ≔ Resending_Channel ∪ {x0 ↦ x1}
act4 : ResendingT_Channel(x0 ↦ x1) ≔ time_channel
END

ReceivingACK ≙
STATUS
ordinary
REFINES
ReceivingACK
ANY
m
x0
x1
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd3 : x0 ∉ dom(ReceivingACK)
grd4 : x0 ↦ x1 ∈ Receiving_Channel ∪ Rereceiving_Channel
grd5 : x0 ↦ x1 ∉ ACKLost
grd6 : Error = FALSE
THEN
act1 : ReceivingACK ≔ ReceivingACK ∪ {x0 ↦ x1}
act2 : transfered ≔ transfered ∪ {m}
act3 : ReceivingACKT(x0 ↦ x1) ≔ time
act4 : ReceivingACK_Channel ≔ ReceivingACK_Channel ∪ {x0 ↦ x1}
act5 : ReceivingACKT_Channel(x0 ↦ x1) ≔ time_channel
END

Error ≙
STATUS
ordinary
REFINES
Error
ANY
x
y
WHERE
grd1 : x ↦ y ∈ Resending
grd2 : x ∉ dom(ReceivingACK)
grd3 : y = MaxResend
grd4 : Error = FALSE
grd5 : time ≥ ResendingT(x ↦ y) + ResendingDL
THEN
act1 : Error ≔ TRUE
END

DataLost ≙
STATUS
ordinary
REFINES
DataLost
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Sending03_Channel ∪ Resending_Channel
grd2 : x0↦x1 ∉ Receiving_Channel ∪ Rereceiving_Channel
grd3 : x0↦x1 ∉ DataLost
THEN
act1 : DataLost ≔ DataLost ∪ {x0 ↦ x1}
END

ACKLost ≙
STATUS
ordinary
REFINES
ACKLost
ANY

x0
x1
WHERE
grd1 : x0↦x1 ∈ Receiving_Channel ∪ Rereceiving_Channel
grd2 : x0↦x1 ∉ ReceivingACK_Channel
grd3 : x0↦x1 ∉ ACKLost
THEN
act1 : ACKLost ≔ ACKLost ∪ {x0 ↦ x1}
END

Receiving ≙
STATUS
ordinary
REFINES
Receiving
ANY
m
x0
x1
y
WHERE
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd3 : x0↦x1 ∈ Resending_Channel ∪ Sending03_Channel
grd4 : x0 ∉ dom(Receiving)
grd5 : x0↦x1 ∉ DataLost
THEN
act1 : Receiving ≔ Receiving ∪ {x0↦x1}
act2 : received ≔ received ∪ {m}
act3 : ReceivingT(x0↦x1) ≔ time_receiver
act4 : Receiving_Channel ≔ Receiving_Channel ∪ {x0↦x1}
act5 : ReceivingT_Channel(x0↦x1) ≔ time_channel
END

Rereceiving ≙
STATUS
ordinary
REFINES
Rereceiving
ANY
x0
x1
WHERE
grd1 : x0↦x1 ∈ Resending_Channel
grd2 : x0↦x1 ∉ DataLost
grd3 : x0↦x1 ∉ Rereceiving
grd4 : x0↦x1 ∉ Receiving
grd5 : x0 ∈ dom(Receiving)
THEN
act1 : Rereceiving ≔ Rereceiving ∪ {x0↦x1}
act2 : RereceivingT(x0↦x1) ≔ time_receiver
act3 : Rereceiving_Channel ≔ Rereceiving_Channel ∪ {x0↦x1}
act4 : RereceivingT_Channel(x0↦x1) ≔ time_channel
END

Receiver_Error ≙
STATUS
ordinary
REFINES
Receiver_Error
ANY
x
y
WHERE
grd1 : Receiver_Error = FALSE
grd2 : Receiving ≠ ∅
grd3 : last ∉ dom(Receiving)
grd4 : x ↦ y ∈ Receiving
grd5 : x < last

grd6 : x+1 ↦ y ∉ Receiving
grd7 : time_receiver ≥ ReceivingT(x ↦ y) + MaxResend ∗ ResendingDL
THEN
act1 : Receiver_Error ≔ TRUE
END

Tick_Tock ≙
STATUS
ordinary
REFINES
Tick_Tock
ANY
tick
WHERE
grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : ∀ x · x↦0 ∈ Sending03 ∧ x↦0 ∉ ReceivingACK ∧ x↦1 ∉ Resending
⇒time+tick≤Sending03T(x↦0)+ResendingDL
grd4 : ∀ x0,x1 · x1 < MaxResend ∧ x0↦x1 ∈ Resending ∧ x0↦x1 ∉ ReceivingACK ∧ x0↦x1+1 ∉ Resending
⇒time+tick≤ResendingT(x0↦x1)+ResendingDL
grd5 : ∀ x0 · x0↦MaxResend ∈ Resending ∧ x0↦MaxResend ∉ ReceivingACK ∧ Error = FALSE
⇒time+tick≤ResendingT(x0↦MaxResend)+ResendingDL
grd6 : ∀ x0,x1 · x0 <last ∧ x0↦x1 ∈ ReceivingACK ∧ x0+1 ↦0 ∉ Sending03 ∧ Error = FALSE
⇒time+tick≤ReceivingACKT(x0↦x1)+0
grd7 : ∀ x · x↦0 ∈ Sending03_Channel ∧ x↦0 ∉ Receiving_Channel ∧ x↦0 ∉ DataLost ⇒ time_channel+tick ≤ Sending03T_Channel(x↦0) + ChannelDL
grd8 : ∀ x,y · x↦y ∈ Resending_Channel ∧ x↦y ∉ Receiving_Channel ∧ x↦y ∉ Rereceiving_Channel ∧ x↦y ∉ DataLost ⇒ time_channel+tick ≤ ResendingT_Channel(x↦y) + ChannelDL
grd9 : ∀ x,y · x↦y ∈ Receiving_Channel ∧ x↦y ∉ ReceivingACK_Channel ∧ x↦y ∉ ACKLost ⇒ time_channel+tick ≤ ReceivingT_Channel(x↦y) + ChannelDL
grd10 : ∀ x,y · x↦y ∈ Rereceiving_Channel ∧ x↦y ∉ ReceivingACK_Channel ∧ x↦y ∉ ACKLost ⇒ time_channel+tick ≤ RereceivingT_Channel(x↦y) + ChannelDL
grd11 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Receiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time_receiver+tick≤ReceivingT(x0↦x1)+ReceiverDL
grd12 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Rereceiving ∧ x0+1 ↦x2 ∉ Receiving ∧x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE
⇒time_receiver+tick≤RereceivingT(x0↦x1)+ReceiverDL
THEN
act1 : time ≔ time+tick
act2 : time_channel ≔ time_channel + tick
act3 : time_receiver ≔ time_receiver + tick
END

END

Decomposed Machine Sender

MACHINE

Sender

SEES

c03

VARIABLES

Sending03
Error
ReceivingACK
transfered
sent2
time
Sending03T
ReceivingACKT
Resending
ResendingT

INVARIANTS

typing_ReceivingACK : ReceivingACK ∈ ℙ(ℤ × ℤ)
typing_Resending : Resending ∈ ℙ(ℤ × ℤ)
typing_time : time ∈ ℤ
typing_Sending03 : Sending03 ∈ ℙ(ℤ × ℤ)
typing_ResendingT : ResendingT ∈ ℙ(ℤ × ℤ × ℤ)
typing_transfered : transfered ∈ ℙ(ℤ × NET_MESSAGE)
typing_sent2 : sent2 ∈ ℙ(ℤ × NET_MESSAGE)
typing_ReceivingACKT : ReceivingACKT ∈ ℙ(ℤ × ℤ × ℤ)
typing_Sending03T : Sending03T ∈ ℙ(ℤ × ℤ × ℤ)
typing_Error : Error ∈ BOOL
m00_inv5 : time∈ℕ
m00_inv7 : dom(Send_Message) = 0 ‥ last
m00_inv8 : last ≥ 0
m01_inv1 : transfered ⊆ Send_Message
m01_inv17 : ∀ x,y · x ∈ ℕ ⇸ NET_MESSAGE ∧ y ⊆ x ∧ dom(x) = dom(y) ⇒ x = y
m03_inv3 : sent2 ⊆ Send_Message
m04_inv4 : Sending03T ∈ Sending03 → ℕ
m06_inv15 : ∀ x · x↦0 ∈ Sending03 ∧ x↦1 ∈ Resending ⇒ ResendingT(x↦1) ≥ Sending03T(x↦0) + ReseendingDL
m06_inv16 : ∀ x,y · x↦y−1 ∈ Resending ∧ x↦y ∈ Resending ⇒ ResendingT(x↦y) ≥ ResendingT(x↦y−1) + ResendingDL

EVENTS

INITIALISATION ≙
STATUS
ordinary

BEGIN

act1 : Sending03 ≔ ∅
act2 : Error ≔ FALSE
act3 : ReceivingACK ≔ ∅
act4 : transfered ≔ ∅
act5 : sent2 ≔ ∅
act6 : time ≔ 0
act7 : Sending03T ≔ ∅
act8 : ReceivingACKT ≔ ∅
act9 : Resending ≔ ∅
act10 : ResendingT ≔ ∅

END

Sending03 ≙
STATUS
ordinary

ANY
m
x
y
WHERE
typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x : x ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
grd3 : x↦0 ∉ Sending03
grd4 : x = 0 ⇒ReceivingACK=∅
grd5 : x > 0 ⇒ x−1 ∈ dom(ReceivingACK)
grd6 : Error = FALSE
THEN
act1 : Sending03 ≔ Sending03 ∪ {x↦0}
act2 : sent2 ≔ sent2 ∪ {m}
act3 : Sending03T(x↦0) ≔ time
END

Resending ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : x0↦x1−1 ∈ Sending03 ∪ Resending
grd2 : x0 ∉ dom(ReceivingACK)
grd3 : x0↦x1 ∉ Resending
grd4 : x1 ≤ MaxResend
grd5 : Error = FALSE
grd6 : x0↦x1−1 ∈ Sending03 ⇒ time ≥ Sending03T(x0↦x1−1) + ResendingDL
grd7 : x0↦x1−1 ∈ Resending ⇒ time ≥ ResendingT(x0↦x1−1) + ResendingDL
THEN
act1 : Resending ≔ Resending ∪ {x0 ↦ x1}
act2 : ResendingT(x0 ↦ x1) ≔ time
END

ReceivingACK ≙
STATUS
ordinary
ANY
m
x0
x1
y
WHERE
typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd3 : x0 ∉ dom(ReceivingACK)
grd6 : Error = FALSE
THEN
act1 : ReceivingACK ≔ ReceivingACK ∪ {x0 ↦ x1}
act2 : transfered ≔ transfered ∪ {m}
act3 : ReceivingACKT(x0 ↦ x1) ≔ time

END

Error ≙
STATUS
ordinary
ANY
x
y
WHERE
typing_y : y ∈ ℤ
typing_x : x ∈ ℤ
grd1 : x ↦ y ∈ Resending
grd2 : x ∉ dom(ReceivingACK)
grd3 : y = MaxResend
grd4 : Error = FALSE
grd5 : time ≥ ResendingT(x ↦ y) + ResendingDL
THEN
act1 : Error ≔ TRUE
END

Tick_Tock ≙
STATUS
ordinary
ANY
tick
WHERE
typing_tick : tick ∈ ℤ
grd1 : tick∈ℕ
grd2 : tick > 0
grd3 : ∀ x · x↦0 ∈ Sending03 ∧ x↦0 ∉ ReceivingACK ∧ x↦1 ∉ Resending
⇒time+tick≤Sending03T(x↦0)+ResendingDL
grd4 : ∀ x0,x1 · x1 < MaxResend ∧ x0↦x1 ∈ Resending ∧ x0↦x1 ∉ ReceivingACK ∧ x0↦x1+1 ∉ Resending
⇒time+tick≤ResendingT(x0↦x1)+ResendingDL
grd5 : ∀ x0 · x0↦MaxResend ∈ Resending ∧ x0↦MaxResend ∉ ReceivingACK ∧ Error = FALSE
⇒time+tick≤ResendingT(x0↦MaxResend)+ResendingDL
grd6 : ∀ x0,x1 · x0 <last ∧ x0↦x1 ∈ ReceivingACK ∧ x0+1 ↦0 ∉ Sending03 ∧ Error = FALSE
⇒time+tick≤ReceivingACKT(x0↦x1)+0
THEN
act1 : time ≔time+tick
END

END

Decomposed Machine Reciever

MACHINE

Receiver

SEES

c03

VARIABLES

Receiving
ReceivingT
received
Rereceiving
RereceivingT
Receiver_Error
time_receiver

INVARIANTS

typing_Receiving : Receiving ∈ ℙ(ℤ × ℤ)
typing_ReceivingT : ReceivingT ∈ ℙ(ℤ × ℤ × ℤ)
typing_time_receiver : time_receiver ∈ ℤ
typing_received : received ∈ ℙ(ℤ × NET_MESSAGE)
typing_Receiver_Error : Receiver_Error ∈ BOOL
typing_RereceivingT : RereceivingT ∈ ℙ(ℤ × ℤ × ℤ)
typing_Rereceiving : Rereceiving ∈ ℙ(ℤ × ℤ)
m00_inv7 : dom(Send_Message) = 0 · · last
m00_inv8 : last ≥ 0
m01_inv17 : ∀ x,y · x ∈ ℕ ⇸ NET_MESSAGE ∧ y ⊆ x ∧ dom(x) = dom(y) ⇒ x = y
m06_inv4 : ReceivingT ∈ Receiving ⟶ ℕ
m06_inv9 : RereceivingT ∈ Rereceiving ⟶ ℕ
m07_inv7 : ∀ x,y · x↦y ∈ Rereceiving ⇒ x ∈ dom(Receiving)

EVENTS

INITIALISATION ≙

STATUS

ordinary

BEGIN

act13 : Receiving ≔ ∅
act14 : ReceivingT ≔ ∅
act15 : received ≔ ∅
act16 : Rereceiving ≔ ∅
act17 : RereceivingT ≔ ∅
act18 : Receiver_Error ≔ FALSE
act29 : time_receiver ≔ 0

END

Receiving ≙

STATUS

ordinary

ANY

m
x0
x1
y

WHERE

typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd4 : x0 ∉ dom(Receiving)

THEN

act1 : Receiving ≔ Receiving ∪ {x0↦x1}
act2 : received ≔ received ∪ {m}

act3 : ReceivingT(x0↦x1) ≔ time_receiver
END

Rereceiving ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd3 : x0↦x1 ∉ Rereceiving
grd4 : x0↦x1 ∉ Receiving
grd5 : x0 ∈ dom(Receiving)
THEN
act1 : Rereceiving ≔ Rereceiving ∪ {x0↦x1}
act2 : RereceivingT(x0↦x1) ≔ time_receiver
END

Receiver_Error ≙
STATUS
ordinary
ANY
x
y
WHERE
typing_y : y ∈ ℤ
typing_x : x ∈ ℤ
grd1 : Receiver_Error = FALSE
grd2 : Receiving ≠ ∅
grd3 : last ∉ dom(Receiving)
grd4 : x ↦ y ∈ Receiving
grd5 : x < last
grd6 : x+1 ↦ y ∉ Receiving
grd7 : time_receiver ≥ ReceivingT(x ↦ y) + MaxResend ∗ ResendingDL
THEN
act1 : Receiver_Error ≔ TRUE
END

Tick_Tock ≙
STATUS
ordinary
ANY
tick
WHERE
typing_tick : tick ∈ ℤ
grd1 : tick∈ℕ
grd2 : tick > 0
grd11 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Receiving ∧ x0+1 ↦x2 ∉ Receiving ∧ x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE ⇒time_receiver+tick≤ReceivingT(x0↦x1)+ReceiverDL
grd12 : ∀ x0,x1,x2 · x0 <last ∧ x0↦x1 ∈ Rereceiving ∧ x0+1 ↦x2 ∉ Receiving ∧ x0 ↦x1+1 ∉ Rereceiving ∧ Receiver_Error = FALSE ⇒time_receiver+tick≤RereceivingT(x0↦x1)+ReceiverDL
THEN
act3 : time_receiver ≔ time_receiver + tick
END

END

Decomposed Machine Channel

MACHINE

Channel

SEES

c03

VARIABLES

DataLost
ACKLost
time_channel
Sending03_Channel
Receiving_Channel
Sending03T_Channel
Resending_Channel
Rereceiving_Channel
ReceivingACK_Channel
ReceivingT_Channel
RereceivingT_Channel
ResendingT_Channel
ReceivingACKT_Channel

INVARIANTS

typing_Resending_Channel : Resending_Channel ∈ ℙ(ℤ × ℤ)
typing_ACKLost : ACKLost ∈ ℙ(ℤ × ℤ)
typing_DataLost : DataLost ∈ ℙ(ℤ × ℤ)
typing_Sending03T_Channel : Sending03T_Channel ∈ ℙ(ℤ × ℤ × ℤ)
typing_RereceivingT_Channel : RereceivingT_Channel ∈ ℙ(ℤ × ℤ × ℤ)
typing_Sending03_Channel : Sending03_Channel ∈ ℙ(ℤ × ℤ)
typing_Rereceiving_Channel : Rereceiving_Channel ∈ ℙ(ℤ × ℤ)
typing_time_channel : time_channel ∈ ℤ
typing_ReceivingACK_Channel : ReceivingACK_Channel ∈ ℙ(ℤ × ℤ)
typing_Receiving_Channel : Receiving_Channel ∈ ℙ(ℤ × ℤ)
typing_ReceivingACKT_Channel : ReceivingACKT_Channel ∈ ℙ(ℤ × ℤ × ℤ)
typing_ReceivingT_Channel : ReceivingT_Channel ∈ ℙ(ℤ × ℤ × ℤ)
typing_ResendingT_Channel : ResendingT_Channel ∈ ℙ(ℤ × ℤ × ℤ)
m00_inv7 : dom(Send_Message) = 0 · · last
m00_inv8 : last ≥ 0
m01_inv17 : ∀ x,y · x ∈ ℕ ⇸ NET_MESSAGE ∧ y ⊆ x ∧ dom(x) = dom(y) ⇒ x = y

EVENTS

INITIALISATION ≙
STATUS
ordinary

BEGIN

act11 : DataLost ≔ ∅
act12 : ACKLost ≔ ∅
act19 : Sending03_Channel ≔ ∅
act20 : Receiving_Channel ≔ ∅
act21 : Sending03T_Channel ≔ ∅
act22 : Resending_Channel ≔ ∅
act23 : Rereceiving_Channel ≔ ∅
act24 : ReceivingACK_Channel ≔ ∅
act25 : ReceivingT_Channel ≔ ∅
act26 : RereceivingT_Channel ≔ ∅
act27 : ResendingT_Channel ≔ ∅
act28 : ReceivingACKT_Channel ≔ ∅
act30 : time_channel ≔ 0

END

Sending03 ≙

STATUS
ordinary
ANY
m
x
y
WHERE
typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x : x ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x ↦ y
THEN
act4 : Sending03_Channel ≔ Sending03_Channel ∪ {x↦0}
act5 : Sending03T_Channel(x↦0) ≔ time_channel
END

Resending ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd4 : x1 ≤ MaxResend
THEN
act3 : Resending_Channel ≔ Resending_Channel ∪ {x0 ↦ x1}
act4 : ResendingT_Channel(x0 ↦ x1) ≔ time_channel
END

ReceivingACK ≙
STATUS
ordinary
ANY
m
x0
x1
y
WHERE
typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd4 : x0 ↦ x1 ∈ Receiving_Channel ∪ Rereceiving_Channel
grd5 : x0 ↦ x1 ∉ ACKLost
THEN
act4 : ReceivingACK_Channel ≔ ReceivingACK_Channel ∪ {x0 ↦ x1}
act5 : ReceivingACKT_Channel(x0 ↦ x1) ≔ time_channel
END

DataLost ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ

typing_x1 : x1 ∈ ℤ
grd1 : x0↦x1 ∈ Sending03_Channel ∪ Resending_Channel
grd2 : x0↦x1 ∉ Receiving_Channel ∪ Rereceiving_Channel
grd3 : x0↦x1 ∉ DataLost
THEN
act1 : DataLost ≔ DataLost ∪ {x0 ↦ x1}
END

ACKLost ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : x0↦x1 ∈ Receiving_Channel ∪ Rereceiving_Channel
grd2 : x0↦x1 ∉ ReceivingACK_Channel
grd3 : x0↦x1 ∉ ACKLost
THEN
act1 : ACKLost ≔ ACKLost ∪ {x0 ↦ x1}
END

Receiving ≙
STATUS
ordinary
ANY
m
x0
x1
y
WHERE
typing_m : m ∈ ℤ × NET_MESSAGE
typing_y : y ∈ NET_MESSAGE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : m ∈ Send_Message
grd2 : m = x0 ↦ y
grd3 : x0↦x1 ∈ Resending_Channel ∪ Sending03_Channel
grd5 : x0↦x1 ∉ DataLost
THEN
act4 : Receiving_Channel ≔ Receiving_Channel ∪ {x0↦x1}
act5 : ReceivingT_Channel(x0↦x1) ≔ time_channel
END

Rereceiving ≙
STATUS
ordinary
ANY
x0
x1
WHERE
typing_x0 : x0 ∈ ℤ
typing_x1 : x1 ∈ ℤ
grd1 : x0↦x1 ∈ Resending_Channel
grd2 : x0↦x1 ∉ DataLost
THEN
act3 : Rereceiving_Channel ≔ Rereceiving_Channel ∪ {x0↦x1}
act4 : RereceivingT_Channel(x0↦x1) ≔ time_channel
END

Tick_Tock ≙
STATUS
ordinary
ANY
tick
WHERE
typing_tick : tick ∈ ℤ
grd1 : tick∈ℕ
grd2 : tick > 0
grd7 : ∀ x · x↦0 ∈ Sending03_Channel ∧ x↦0 ∉ Receiving_Channel ∧ x↦0 ∉ DataLost ⇒ time_channel+tick ≤ Sending03T_Channel(x↦0) + ChannelDL
grd8 : ∀ x,y · x↦y ∈ Resending_Channel ∧ x↦y ∉ Receiving_Channel ∧ x↦y ∉ Rereceiving_Channel ∧ x↦y ∉ DataLost ⇒ time_channel+tick ≤ ResendingT_Channel(x↦y) + ChannelDL
grd9 : ∀ x,y · x↦y ∈ Receiving_Channel ∧ x↦y ∉ ReceivingACK_Channel ∧ x↦y ∉ ACKLost ⇒ time_channel+tick ≤ ReceivingT_Channel(x↦y) + ChannelDL
grd10 : ∀ x,y · x↦y ∈ Rereceiving_Channel ∧ x↦y ∉ ReceivingACK_Channel ∧ x↦y ∉ ACKLost ⇒ time_channel+tick ≤ RereceivingT_Channel(x↦y) + ChannelDL
THEN
act2 : time_channel ≔ time_channel + tick
END

END