**Process Injection/Communication Ring Stretching Algorithm INJ**

(Generalized Formal Specification[[1]](#footnote-1))

**Pi::INJ**

**{SYNOPSIS}**

The algorithm is used to automatically insert (or inject) a new process into communication ring i.e. stretching the ring.

The process injected might be one of spare (backup) processes or a repaired process removed off the ring some time ago because of its failure.

Failures of other process are acceptable during the operation.

Positive side effect is fixing of ring integrity from the injected process at the end of the algorithm.

**{ASSUMPTIONS}**

The INJ algorithm is valid under next conditions:

* The distributed system is synchronous.
* The type of process failures is “fail-stop”.
* The injected process should not duplicate a process already part of the ring.

**ALGORITHM 1**: Declarative Part of *Pi::INJ*

**{SYSTEM CONSTANTS}**

Int MAX\_INJ\_PERIOD // period between injection attempts

PId i // process Pi identifier

**{MESSAGES}**

<mrk\_inj, j, k>

**{SET OF STATES}**

<State> := {OFF, ON}

**{INTERNAL STATE SPACE}**

State state // current process INJ state

RUP::state // current process RUP state

Timer TimerINJ // timer

CEH::ListPIds // list of process identifiers

CEH::PIdNext // current neighbor process identifier

**ALGORITHM 2**: Event Handlers of *Pi::INJ*

**OnInit:**

state := ON

RUP::state := DOWN

TimerINJ.Interval := MAX\_INJ\_PERIOD

CEH::ListPIds.Set()

CEH::PIdNext := CEH::ListPIds.Front() // The Head is not popped out

ChannelOut.Open()

**OnOutputConnect:**

*{Injection First Attempt}*

OnStart()

**OnStart:**

Send <mrk\_inj, i, pidNext>

TimerINJ.Start()

**OnReceiptOf <mrk\_inj, j, k>:**

**If** i = j

*{Injection End}*

TimerINJ.Stop()

state := OFF

RUP::state := UP

*{Distributed Election Entry Point}*

E::OnStartElection()

**Else**

**If** j ∉ CEH::ListPIds

CEH::ListPIds.Insert(j, k)

**EndIf**

**If** CEH::pidNext == k

*{Output Channel Reconnect}*

ChannelOut.Close()

CEH::PIdNext := CEH::ListPIds.Front()

ChannelOut.Open()

**EndIf**

Send <mrk\_inj, j, k>

*{Start Ring Check Up Algorithm}*

RUP::OnStart()

**EndIf**

**OnTimer:**

*{Injection Next Attempt}*

OnStart()

1. Implementation <https://github.com/milphaser/XME.Ring> [↑](#footnote-ref-1)