RSTP STATE MACHINES - AN OVERVIEW

NOTE: THIS OVERVIEW IS NOT ITSELF A STATE MACHINE BUT SERVES TO ILLUSTRATE THE PRINCIPAL VARIABLES THAT ARE USED TO COMMUNICATE BETWEEN THE INDIVIDUAL RSTP STATE MACHINES AND THE VARIABLES LOCAL TO EACH MACHINE.

ABBREVIATIONS:
- PIM: Port Information Machine
- PRS: Port Role Selection Machine
- PRT: Port Role Transition Machine
- PST: Port State Transition Machine
- RCM: Role Confirmation Machine
- TCM: Topology Change Machine
- PPM: Port Protocol Migration Machine
- PRTI: Port Role Information State Machine
- PTX: Port Transmit Machine
- PTI: Port Timers Machine

NOTE: For convenience all timers are collected together into one state machine.
selectDisabled(bridge);
/* Sets selectedRole to DisabledPort for all bridge ports. */

selectRoles(bridge);
/* Assigns a role, i.e. sets selectedRole to one of DisabledPort, RootPort, DesignatedPort, AlternatePort, or BackupPort for each port of the bridge.
First selects the RootPort from ports with (portIs == Received), choosing the port with the best Spanning Tree information after the port's Path Cost has been added. If this Spanning Tree information differs from that held as rootInfo for the bridge, updates the latter and clears inSync and sync for all bridge ports.
Then sets updtInfo for all other ports for which:
- a) (portIs == Received) and the Spanning Tree Information, without the port's own Path Cost added, is worse than the RootPort's with its Path Cost added
- b) (portIs == Aged)
- c) (portIs == Mine) and the Spanning Tree Information for the port or the associated timer parameters differ from those with the for the RootPort with its Path Cost added.

clearReselect(bridge);
/* Sets reselect = FALSE for all ports */

PRT: PORT ROLE TRANSITIONS STATE MACHINE (PER PORT)

PAGE 2 OF 3 : SHOWING THE ROOT PORT STATES

NOTE: All transitions, except UCT, qualified by "&& !reselect".

reRooted = ((rrWhile1 == 0) && (rrWhile2 == 0) && ... (rrWhileN == 0)) for all ports except this Root Port
synced = (sync1 && sync2 && ... syncn) for all ports except this Root Port

syncBridge()

/* Sets syncPort for all other ports. */
}
}

reRootBridge()

/* Sets reRoot for all other ports. */
}
PRT: PORT ROLE TRANSITIONS STATE MACHINE (PER PORT)

NOTE: All transitions, except UCT, qualified by "&& !reselect".
NOTE: A small system dependent delay may occur on each of the transitions shown.
**TCM: TOPOLOGY CHANGE MACHINE (PER PORT)**

**TOPOLOGY CHANGE DETECTION, NOTIFICATION, PROPAGATION AND FILTERING DATABASE FLUSHING**

BEGIN

- **INIT**
  - \(\text{flush}(); \text{tcWhile} = 0; \text{tc} = \text{tcProp} = \text{FALSE};;\)

- **ACTIVE**
  - (role != Root::port) && (role != Designated::port)

- **INACTIVE**
  - rcvdTc = rcvdTcn = rcvdTcAck = tc = tcProp = FALSE;

- **NOTIFIED**
  - if Designated tcAck = TRUE; 
    - rcvdTcn = FALSE; rcvdTc = FALSE; 
    - tcPropagation();

- **DETECTED**
  - tcWhile = 2*Hello_time; 
    - tcPropagation()

- **ACKNOWLEDGED**
  - tcWhile = 0; 
    - flush();

- **PROPAGATING**
  - rcvdTc = rcvdTcn = rcvdTcAck = tc = tcProp = FALSE;

- **UCT**
  - (role == Root::port) || (role == Designated::port)

- (role != Root::port) && (role != Designated::port)

\(tc\text{Propagation}()\)
\{ /* Sets \text{tcprop} for all other ports. */ \}
\}
\)
\(\text{flush}()\)
\{ /* Flushes the filtering database for this port. !!! Unless an edge port!!! */ \}
\)}
PORT TIMERS STATE MACHINE (PER PORT)

BEGIN

tick == TRUE

dec(x)

| if (x != 0) x = x - 1; |

}
msyncWhile = MigrateSync;
mcheck = FALSE;
sendNew = TRUE;
SEND_OLD
((msyncWhile == 0) && rcvdOld)
BEGIN || (portInfo == Disabled)
SENDING_NEW
msyncWhile = MigrateSync;
rcheck = FALSE;
SENDING_NEW
msyncWhile = MigrateSync;
rcheck = FALSE;
SEND OLD
((msyncWhile != 0) && (rcvdOld || rcvdNew)
rcvdNew = rcvdOld = FALSE;
SEND OLD
rcvdNew = rcvdOld = FALSE;
SENDING OLD
(((msyncWhile == 0) && rcvdOld)
rcvdNew = rcvdOld = FALSE;
SENDING OLD
rcvdNew = rcvdOld = FALSE;
((msyncWhile != 0) && rcvdNew)
rcvdNew = rcvdOld = FALSE;
SENDING OLD
rcvdNew = rcvdOld = FALSE;
rcvdNew = rcvdOld = FALSE;
mscheck
mscheck
mcheck
mcheck
PTX: PORT TRANSMIT MACHINE (PER PORT)

BEGIN

INIT
helloWhen = 0;
txCnt = 0;

UCT

(IDLE)
newInfo = FALSE;

sendNew & (txCount < TxHoldCount) &
((role == Root_port) & (helloWhen == 0) || newInfo)) ||
((role == Designated_port) & (helloWhen == 0) & (tcWhile != 0)) || newInfo)

(sendNew & (txCount < TxHoldCount) &
((role == Root_port) & (helloWhen == 0) || newInfo) & (tcWhile != 0)) || newInfo)

!sendNew & (txCount < TxHoldCount) &
((role == Designated_port) & (helloWhen == 0) || newInfo)

txRstp();

txCount += 1;

tcAck = FALSE;

helloWhen = HelloTime;

UCT

(INIT)

helloWhen = 0;
txCnt += 1;

UCT

TRANSMIT_RSTP

 txRstp();
txCount = 1;
tcAck = FALSE;

helloWhen = HelloTime;

UCT

TRANSMIT_TCN

 txTcn();
txCount += 1;

tcAck = FALSE;

helloWhen = HelloTime;

UCT

TRANSMIT_CONFIG

 txConfig();
txCount += 1;

tcAck = FALSE;

helloWhen = HelloTime;

UCT

PTX: PORT TRANSMIT MACHINE (PER PORT)

BEGIN

INIT
helloWhen = 0;
txCnt = 0;

UCT

(IDLE)
newInfo = FALSE;

sendNew & (txCount < TxHoldCount) &
((role == Root_port) & (helloWhen == 0) || newInfo)) ||
((role == Designated_port) & (helloWhen == 0) & (tcWhile != 0)) || newInfo)

(!sendNew & (txCount < TxHoldCount) &
((role == Root_port) & (helloWhen == 0) || newInfo) & (tcWhile != 0)) || newInfo)

!sendNew & (txCount < TxHoldCount) &
((role == Designated_port) & (helloWhen == 0) || newInfo)

txRstp()

(^ Transmits an RSTP BPDU. The Root and Root Path Cost parameters in the BPDU are set as stored for rootInfo for the transmitting bridge.
The Designated Bridge and Designated Port parameters are set as for the transmitting bridge. The port role flags are set to the role of
the transmitting port and the inSync and synReq flags are set to the values of the sync and reqSync flags for the transmitting port respectively. The
topology change flag is set if tcWhile != 0 for the port. The topology change notification and topology change acknowledge flags in the BPDU
are never used (propose that topology change notification flag not be added as previously suggested). )

) txRstp()

(txConfig)

(^ Transmits an Config BPDU. The Root and Root Path Cost parameters in the BPDU are set as stored for rootInfo for the transmitting bridge.
The Designated Bridge and Designated Port parameters are set as for the transmitting bridge. The topology change flag is set if tcWhile != 0 for
the port. The topology change notification flag is set to the value of TcAck for the port. )

) txConfig()
NOTATION:
In the Overview variables are shown both within the machine where they are principally used and between machines where they are
use to communicate information. In the latter case they are shown with a variety of arrow styles, running from one machine to another,
that provide an overview of how the variables are typically used:

Not changed by the target machine. Where the state machines are both per port, this variable communicates between machine
instances for the same port.

Set (or cleared) by the originating machine, clear (or set) by the target machine. Where the state machines are both per port, this
variable communicates between machine instances for the same port.

As above except that the originating per port machine instance communicates with multiple port machine instances (by setting or
clearing variables owned by those ports).

As above except that multiple per port instances communicate with (an)other instance(s) (by setting or clearing variables owned by
the originating ports).

typedef enum {BetterDesignatedMsg, RepeatedDesignatedMsg, OtherMsg} RcvdMsg;
typedef enum {Disabled, Mine, Received, Aged} InfoIs;
typedef struct /* StpInfo */
{
  Priority root_pri;
  Bridge_id root_id;
  Stp_cost root_cost;
  Priority bridge_pri;
  Bridge_id bridge_id;
  Priority port_pri;
  Port_id port_id;
  Centisecs message_age;
  Centisecs max_age;
  Centisecs forward_delay;
  Centisecs hello_time;
}