OVERVIEW

ABBREVIATIONS:
PIM: Port Information Machine
PRS: Port Role Selection Machine
PRT: Port Role Transitions Machine
PST: Port State Transitions Machine
TCM: Topology Change Machine
PPM: Port Protocol Migration Machine
PTX: Port Transmit Machine
PTI: Port Timers Machine

NOTE 1: For convenience all timers are collected together into one state machine.

NOTE 2: This overview diagram is not itself a state machine, but serves to illustrate the principal variables that are used to communicate between the individual RST state machines and the variables local to each machine.

NOTATION:
Variables are shown both within the machine where they are principally used and between machines where they are used 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, cleared (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 (another instance/s) (by setting or clearing variables owned by the originating Ports).
PORT ROLE SELECTION

BEGIN

INIT_BRIDGE

updRoleDisabledBridge();
rootInfo = rootPriority();

ROLE_SELECTION

clearReselectBridge();
updRolesBridge();
setSelectedBridge();
reselect1 || reselect2 || ... || reselectN

JCT
DISABLED, ALTERNATE, BACKUP

BEGIN

INIT_PORT
role = DisabledPort;
synced = FALSE;
sync = reRoot = TRUE;
\text{rWhile} = \text{fdWhile} = \text{FwdDelay};
rbWhile = 0;

UCT

BLOCK_PORT
role = selectedRole;
learn = forward = FALSE;

UCT

BACKUP_PORT
rbWhile = 2*HelloTime;

UCT

BLOCKED_PORT
fdWhile = FwdDelay;
synced = TRUE; rWhile = 0;
sync = reRoot = FALSE

((fdWhile \neq FwdDelay) \lor synced \lor reRoot \lor \neg synced)

All transitions, except UCT, are qualified by "&& selected \&\& !updtInfo".
All transitions, except UCT, are qualified by "&& selected && !updateInfo".
The following abbreviations are used in this diagram:

allSynced: (agree1 && agree2 && ... && agreeN) for all Ports other than this Root Port.

reRooted: (rWhile1 == 0) && (rWhite2 == 0) && ... && (rWhileN == 0) for all ports except this Root Port.
DESIGNATED

All transitions, except UCT, are qualified by "&& selected && updtInfo."
PORT STATE TRANSITIONS

BEGIN

DISCARDING
disableLearning(); learning = FALSE;
disableForwarding(); forwarding = FALSE;

LEARNING
enableLearning();
learning = TRUE;

FORWARDING
tc = topoEdge;
enableForwarding();
forwarding = TRUE;

NOTE: A small system dependent delay may occur on each of the transitions shown.
PORT TIMERS

dec(HelloWhen); dec(tcWhile); dec(fdWhile); dec(infoAge);
dec(nWhite); dec(rbWhile); dec(msyncWhile);
tick = FALSE;

TICK $\rightarrow$ BEGIN $\rightarrow$ ONE_SECOND $\rightarrow$ Tick $\equiv$ TRUE

The following abbreviation is used in this state diagram:
dec(x)
\{ if (x \neq 0) x = x - 1; \}
PORT PROTOCOL MIGRATION

BEGIN || !portEnabled

SEND RSTP
mdelayWhile = MigrateTime;
mcheck = FALSE;
sendRSTP = TRUE;

SENDING_RSTP
rcvdRSTP = rcvdSTP = FALSE;

SEND STP
mdelayWhile = MigrateTime;
sendRSTP = FALSE;

SENDING_STP
rcvdRSTP = rcvdSTP = FALSE;

((mdelayWhile == 0) && rcvdSTP) || (forceVersion < 2)

((mdelayWhile != 0) && (rcvdSTP || rcvdRSTP))

((mdelayWhile == 0) && rcvdSTP) && (forceVersion >= 2)
PORT TRANSMIT

BEGIN

INIT
    newInfo = FALSE;
    helloWhen = 0;
    txCount = 0;
    txTcn(); txCount +=1;
    helloWhen = HelloTime;

TRANSMIT_CONFIG
    newInfo = FALSE;
    txTcn(); txCount +=1;
    tcAck = FALSE;
    helloWhen = HelloTime;

RLSE
    txCount = 0;

TRANSMIT_TCN
    newInfo = FALSE;
    txTcn(); txCount +=1;
    tcAck = FALSE;
    helloWhen = HelloTime;

TRANSMIT_RSTP
    newInfo = FALSE;
    txRstp(); txCount +=1;
    tcAck = FALSE;
    helloWhen = HelloTime;

IDLE

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

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

sendNew && (txCount < TxHoldCount) &&
    ((role == DesignatedPort) && (helloWhen == 0))

sendNew && (txCount < TxHoldCount) &&
    ((role == RootPort) && (helloWhen == 0))

sendNew && (txCount < TxHoldCount) &&
    ((role == DesignatedPort) && (helloWhen == 0))

All transitions, except UCT, are qualified by "&& selected &&!updtInfo".