```
──────────────────────── MODULE RelayServer ────────────────────────

EXTENDS Integers

CONSTANT
PARTIES,   The set of parties, i.e p1,p2,p3
ROUNDS  The set of rounds, i.e 1,2,3,4

ASSUME ROUNDS ⊆ Nat

VARIABLES
partyState,   partyState[p] is the state of party r.
serverState,   The state of the server.
readyParties,   The set of parties that signal they are ready
assignedParties,   The set of parties that the server assigned them ID
msgs

────────────────────────────────────────────────────────────────────

Messages ≜
[type : {"Abort", "Start"}]
∪ [type : {"Ready"}, party : PARTIES]
∪ [type : {"Assign"}, party : PARTIES]
∪ [type : {"AbortReq"}, party : PARTIES]

∪ [type : {"P2P"}, from : PARTIES, to : PARTIES, round : ROUNDS \ {0}]
∪ [type : {"RelayP2P"}, from : PARTIES, to : PARTIES, round : ROUNDS \ {0}]
∪ [type : {"Broadcast"}, party : PARTIES, round : ROUNDS \ {0}]
∪ [type : {"RelayBroadcast"}, party : PARTIES, round : ROUNDS \ {0}]

────────────────────────────────────────────────────────────────────

TypeOK ≜
∧ partyState ∈ [PARTIES → {"idle", "ready", "assigned", "aborted"}]
∧ serverState ∈ {"init", "running"}
∧ readyParties ⊆ PARTIES
∧ assignedParties ⊆ PARTIES
```

$\land\ msgs \subseteq Messages$

$Init\ \triangleq$
$\land\ partyState = [p \in PARTIES \mapsto \text{“idle”}]$
$\land\ serverState = \text{“init”}$
$\land\ readyParties = \{\}$
$\land\ assignedParties = \{\}$
$\land\ msgs = \{\}$

$PartyReady(p)\ \triangleq$
$\land\ serverState = \text{“init”}$
$\land\ msgs' = msgs \cup \{[type \mapsto \text{“Ready”},\ party \mapsto p]\}$
$\land\ p \notin readyParties$
$\land\ readyParties' = readyParties \cup \{p\}$
$\land\ partyState' = [partyState\ \text{EXCEPT}\ ![p] = \text{“ready”}]$
$\land\ \text{UNCHANGED}\ \langle serverState,\ assignedParties \rangle$

$Assign(p)\ \triangleq$
$\land\ serverState = \text{“init”}$
$\land\ [type \mapsto \text{“Ready”},\ party \mapsto p] \in msgs$
$\land\ [type \mapsto \text{“Assign”},\ party \mapsto p] \notin msgs$
$\land\ msgs' = msgs \cup \{[type \mapsto \text{“Assign”},\ party \mapsto p]\}$
$\land\ assignedParties' = assignedParties \cup \{p\}$
$\land\ partyState' = [partyState\ \text{EXCEPT}\ ![p] = \text{“assigned”}]$
$\land\ \text{UNCHANGED}\ \langle serverState,\ readyParties \rangle$

$Start\ \triangleq$
$\land\ serverState = \text{“init”}$
$\land\ assignedParties = PARTIES$
$\land\ serverState' = \text{“running”}$
$\land\ msgs' = msgs \cup \{[type \mapsto \text{“Start”}]\}$
$\land\ \text{UNCHANGED}\ \langle partyState,\ readyParties,\ assignedParties \rangle$

$PartyAbort(p)\ \triangleq$
$\land\ partyState[p] = \text{“assigned”}$

$$\begin{array}{l}
\land\ \mathit{partyState}' = [\mathit{partyState}\ \textsc{except}\ ![p] = \textsf{“aborted”}]\\
\land\ [\mathit{type} \mapsto \textsf{“AbortReq”},\ \mathit{party} \mapsto p] \in \mathit{msgs}\\
\land\ \mathit{serverState} = \textsf{“running”}\\
\land\ \mathit{serverState}' = \textsf{“init”}\\
\land\ \mathit{msgs}' = \{[\mathit{type} \mapsto \textsf{“Abort”}]\}\\
\land\ \textsc{unchanged}\ \langle \mathit{readyParties} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{Abort}\ \triangleq\\
\land\ \mathit{serverState} = \textsf{“init”}\\
\land\ [\mathit{type} \mapsto \textsf{“Abort”}] \in \mathit{msgs}\\
\land\ \mathit{readyParties}' = \{\}\\
\land\ \mathit{msgs}' = \{\}\\
\land\ \mathit{partyState}' = [p \in \mathit{PARTIES} \mapsto \textsf{“idle”}]\\
\land\ \textsc{unchanged}\ \langle \mathit{serverState} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{ReqToBroadcast}(r,\ p)\ \triangleq\\
\land\ \mathit{assignedParties} = \mathit{PARTIES}\\
\land\ \mathit{serverState} = \textsf{“running”}\\
\land\ [\mathit{type} \mapsto \textsf{“Broadcast”},\ \mathit{party} \mapsto p,\ \mathit{round} \mapsto r] \notin \mathit{msgs}\\
\land\ \mathit{msgs}' = \mathit{msgs} \cup \{[\mathit{type} \mapsto \textsf{“Broadcast”},\ \mathit{party} \mapsto p,\ \mathit{round} \mapsto r]\}\\
\land\ \textsc{unchanged}\ \langle \mathit{serverState},\ \mathit{partyState},\ \mathit{readyParties},\ \mathit{assignedParties} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{RelayBroadcast}(r,\ p)\ \triangleq\\
\land\ \mathit{assignedParties} = \mathit{PARTIES}\\
\land\ \mathit{serverState} = \textsf{“running”}\\
\land\ [\mathit{type} \mapsto \textsf{“Broadcast”},\ \mathit{party} \mapsto p,\ \mathit{round} \mapsto r] \in \mathit{msgs}\\
\land\ [\mathit{type} \mapsto \textsf{“RelayBroadcast”},\ \mathit{party} \mapsto p,\ \mathit{round} \mapsto r] \notin \mathit{msgs}\\
\land\ \mathit{msgs}' = \mathit{msgs} \cup \{[\mathit{type} \mapsto \textsf{“RelayBroadcast”},\ \mathit{party} \mapsto p,\ \mathit{round} \mapsto r]\}\\
\land\ \textsc{unchanged}\ \langle \mathit{serverState},\ \mathit{partyState},\ \mathit{readyParties},\ \mathit{assignedParties} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{ReqToP2P}(r,\ p1,\ p2)\ \triangleq\\
\land\ \mathit{assignedParties} = \mathit{PARTIES}\\
\land\ \mathit{serverState} = \textsf{“running”}\\
\land\ [\mathit{type} \mapsto \textsf{“P2P”},\ \mathit{from} \mapsto p1,\ \mathit{to} \mapsto p2,\ \mathit{round} \mapsto r] \notin \mathit{msgs}\\
\land\ \mathit{msgs}' = \mathit{msgs} \cup \{[\mathit{type} \mapsto \textsf{“P2P”},\ \mathit{from} \mapsto p1,\ \mathit{to} \mapsto p2,\ \mathit{round} \mapsto r]\}
\end{array}$$

$$\begin{array}{l}
\land \textsc{unchanged}\ \langle \mathit{serverState},\ \mathit{partyState},\ \mathit{readyParties},\ \mathit{assignedParties} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{RelayP2P}(r,\ p1,\ p2)\ \triangleq \\
\land\ \mathit{assignedParties} = \mathit{PARTIES} \\
\land\ \mathit{serverState} = \text{“running”} \\
\land\ [\mathit{type} \mapsto \text{“P2P”},\ \mathit{from} \mapsto p1,\ \mathit{to} \mapsto p2,\ \mathit{round} \mapsto r] \in \mathit{msgs} \\
\land\ [\mathit{type} \mapsto \text{“RelayP2P”},\ \mathit{from} \mapsto p1,\ \mathit{to} \mapsto p2,\ \mathit{round} \mapsto r] \notin \mathit{msgs} \\
\land\ \mathit{msgs}' = \mathit{msgs} \cup \{[\mathit{type} \mapsto \text{“RelayP2P”},\ \mathit{from} \mapsto p1,\ \mathit{to} \mapsto p2,\ \mathit{round} \mapsto r]\} \\
\land\ \textsc{unchanged}\ \langle \mathit{serverState},\ \mathit{partyState},\ \mathit{readyParties},\ \mathit{assignedParties} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{Next}\ \triangleq \\
\mathit{Start} \lor \mathit{Abort} \\
\lor\ (\exists\, p \in \mathit{PARTIES} : \mathit{PartyAbort}(p)) \\
\lor\ (\exists\, p \in \mathit{PARTIES} : \mathit{PartyReady}(p)) \\
\lor\ (\exists\, p \in \mathit{PARTIES} : \mathit{Assign}(p)) \\
\lor\ (\exists\, p \in \mathit{PARTIES} : \exists\, r \in \mathit{ROUNDS} : \mathit{ReqToBroadcast}(r,\ p)) \\
\lor\ (\exists\, p \in \mathit{PARTIES} : \exists\, r \in \mathit{ROUNDS} : \mathit{RelayBroadcast}(r,\ p)) \\
\lor\ (\exists\, p1 \in \mathit{PARTIES} : \exists\, p2 \in \mathit{PARTIES} : \exists\, r \in \mathit{ROUNDS} : \mathit{ReqToP2P}(r,\ p1,\ p2)) \\
\lor\ (\exists\, p1 \in \mathit{PARTIES} : \exists\, p2 \in \mathit{PARTIES} : \exists\, r \in \mathit{ROUNDS} : \mathit{RelayP2P}(r,\ p1,\ p2))
\end{array}$$

$$\begin{array}{l}
\mathit{Spec}\ \triangleq\ \mathit{Init} \land \Box[\mathit{Next}]_{\langle \mathit{partyState},\, \mathit{serverState},\, \mathit{readyParties},\, \mathit{assignedParties},\, \mathit{msgs} \rangle} \\
\textsc{theorem}\ \mathit{Spec} \Rightarrow \Box\, \mathit{TypeOK}
\end{array}$$