MODULE *Nomination*

This is a high-level specification of *SCP* focusing on the nomination protocol.

Currently, as implemented, before voting for a txset hash, nodes wait to obtain its preimage. Delaying the point at which we wait for the pre-image would leave more room for disseminating the txset in parallel to nomination. However this has to be done carefully to maintain the main property of nomination: assuming that there is a nomination round with a good leader and during which the network is fast engough, at least a Tier-1 quorum must eventually enter balloting.

In the version specified in this document, we do not wait on the pre-image to vote for a txset hash, but we do wait for the pre-image before accepting it.

In the previous version of this document, we even accepted without a pre-image. There is a problem with this: it could create a situation in which not enough nodes can start balloting ($i.e.$ not a full quorum) and the whole system is stuck.

The problem stems from the fact that, in the nomination protocol, nodes that confirm a candidate then stop voting for new values (otherwise nomination is not guaranteed to converge). So if a blocking set $B$ confirms a candidate but somehow other nodes cannot get the pre-images they need to do so, more nomination rounds will not help because the members of $B$ have stopped voting, which blocks the progress of any new candidate. Depending on how pre-images are disseminated, this can potentially be exploited by an attacker to halt the system.

So accepting without a pre-image is only workable if there is some way to guarantee that, once a Tier-1 blocking set has a pre-image, then everybody in Tier-1 eventually gets it.

Another problem is that we want it to be likely that a quorum starts balloting already in agreement and roughly at the same time. If we delay checking pre-images to the confirm stage, an attacker could first send the pre-image to a set A of nodes, which then enter balloting at time $T\_A$, but not send the pre-image to another set $B$ of nodes, which then enter balloting at time $T\_B \rangle T\_A$ because they need to get the pre-image from A before starting balloting. For example, if it takes $500ms$ for members of $B$ to get the pre-image from members of A, then $T\_B = T\_A + 500ms$. This can cause the first ballot to end without a decision. Members of $B$ could also start a new nomination round before $T\_B$ and then enter balloting not only late but also with a different value than members of A.

EXTENDS *Naturals*, *FiniteSets*

CONSTANTS
- $V$, validators
- $TxSet$, blocks
- $Bot$, default value
- $Quorum(\_)$, $Quorum(v)$ is the set of quorums of validator $v$
- $Blocking(\_)$, $Blocking(v)$ is the set of blocking sets of validator $v$
- $Combine(\_)$, the functions that combines candidates to produce a txset for balloting
- $H$, domain of hashes
- $Hash(\_)$ hash function

VARIABLES *txSetForBalloting*, *voted*, *accepted*, *round*, *candidates*, *preImage*, *leader*

$vars \triangleq \langle txSetForBalloting, voted, accepted, round, candidates, preImage, leader \rangle$

$$
\begin{array}{l}
\mathit{Init} \triangleq \\
\quad \land\ \mathit{txSetForBalloting} = [v \in V \mapsto \mathit{Bot}] \\
\quad \land\ \mathit{voted} = [v \in V \mapsto \{\}] \ \text{variable } X \text{ in the whitepaper} \\
\quad \land\ \mathit{accepted} = [v \in V \mapsto \{\}] \ \text{variable } Y \text{ in the whitepaper} \\
\quad \land\ \mathit{round} = [v \in V \mapsto 0] \\
\quad \land\ \mathit{candidates} = [v \in V \mapsto \{\}] \ \text{variable } Z \text{ in the whitepaper} \\
\quad \land\ \mathit{preImage} = [v \in V \mapsto [h \in H \mapsto \mathit{Bot}]] \\
\quad \land\ \mathit{leader} = [v \in V \mapsto \mathit{Bot}]
\end{array}
$$

$$
\begin{array}{l}
\mathit{StartRound}(v) \triangleq \\
\quad \land\ \mathit{round}' = [\mathit{round} \ \text{EXCEPT} \ ![v] = \mathit{round}[v] + 1] \\
\quad \land\ \exists\, l \in V : \\
\qquad \land\ \mathit{leader}' = [\mathit{leader} \ \text{EXCEPT} \ ![v] = l] \\
\qquad \land\ \text{IF} \ l = v \\
\qquad\quad \text{THEN} \ \exists\, \mathit{txs} \in \mathit{TxSet} : \\
\qquad\qquad \land\ \mathit{preImage}' = [\mathit{preImage} \ \text{EXCEPT} \ ![v][\mathit{Hash}(\mathit{txs})] = \mathit{txs}] \\
\qquad\qquad \land\ \mathit{voted}' = [\mathit{voted} \ \text{EXCEPT} \ ![v] = \mathit{voted}[v] \cup \{\mathit{Hash}(\mathit{txs})\}] \\
\qquad\quad \text{ELSE} \ \text{UNCHANGED} \ \langle \mathit{voted}, \mathit{preImage} \rangle \\
\quad \land\ \text{UNCHANGED} \ \langle \mathit{txSetForBalloting}, \mathit{accepted}, \mathit{candidates} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{Vote}(v) \triangleq \\
\quad \land\ \text{IF} \ \mathit{candidates}[v] = \{\} \\
\qquad \text{THEN} \ \land\ \mathit{leader}[v] \neq \mathit{Bot} \\
\qquad\quad \land\ \text{LET} \ \mathit{hs} \triangleq \mathit{voted}[\mathit{leader}[v]] \ \text{IN} \\
\qquad\qquad \land\ \mathit{hs} \neq \{\} \\
\qquad\qquad \land\ \mathit{voted}' = [\mathit{voted} \ \text{EXCEPT} \ ![v] = \mathit{voted}[v] \cup \mathit{hs}] \\
\qquad \text{ELSE} \ \text{UNCHANGED} \ \mathit{voted} \\
\quad \land\ \text{UNCHANGED} \ \langle \mathit{txSetForBalloting}, \mathit{accepted}, \mathit{round}, \mathit{candidates}, \mathit{preImage}, \mathit{leader} \rangle
\end{array}
$$

$$
\mathit{VotedHashes} \triangleq \text{UNION} \ \{\mathit{voted}[v] : v \in V\}
$$

$$
\begin{array}{l}
\mathit{GetTxSet}(v, \mathit{txs}) \triangleq \\
\quad \land\ \mathit{Hash}(\mathit{txs}) \in \mathit{VotedHashes} \\
\quad \land\ \mathit{preImage}' = [\mathit{preImage} \ \text{EXCEPT} \ ![v][\mathit{Hash}(\mathit{txs})] = \mathit{txs}] \\
\quad \land\ \text{UNCHANGED} \ \langle \mathit{txSetForBalloting}, \mathit{voted}, \mathit{accepted}, \mathit{round}, \mathit{candidates}, \mathit{leader} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{Accept}(v, h) \triangleq \\
\quad \land\ \mathit{preImage}[v][h] \neq \mathit{Bot} \\
\quad \land\ \lor \exists\, Q \in \mathit{Quorum}(v) : \forall\, w \in Q : h \in \mathit{voted}[w] \lor h \in \mathit{accepted}[w] \\
\qquad \lor \exists\, \mathit{Bl} \in \mathit{Blocking}(v) : \forall\, w \in \mathit{Bl} : h \in \mathit{accepted}[w] \\
\quad \land\ \mathit{accepted}' = [\mathit{accepted} \ \text{EXCEPT} \ ![v] = \mathit{accepted}[v] \cup \{h\}] \\
\quad \land\ \text{UNCHANGED} \ \langle \mathit{txSetForBalloting}, \mathit{voted}, \mathit{round}, \mathit{candidates}, \mathit{preImage}, \mathit{leader} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{Confirm}(v, h) \triangleq \exists\, Q \in \mathit{Quorum}(v) : \\
\quad \land\ \ \mathit{preImage}[v][h] \neq \mathit{Bot} \\
\quad \land\ \ \forall\, w \in Q : h \in \mathit{accepted}[w] \\
\quad \land\ \ \mathit{candidates}' = [\mathit{candidates} \ \text{EXCEPT} \ ![v] = \mathit{candidates}[v] \cup \{\mathit{preImage}[v][h]\}]
\end{array}
$$

$\land \quad txSetForBalloting' = [txSetForBalloting \text{ EXCEPT } ![v] = Combine(candidates'[v])]$
$\land \quad \text{UNCHANGED } \langle voted,\ accepted,\ round,\ preImage,\ leader \rangle$

$Next \triangleq \exists\, v \in V,\ txs \in TxSet,\ h \in H :$
$\lor\ StartRound(v)$
$\lor\ Vote(v)$
$\lor\ GetTxSet(v,\ txs)$
$\lor\ Accept(v,\ h)$
$\lor\ Confirm(v,\ h)$

Here we assume that all agree on a leader in round 3 and stay in round 3 forever (for liveness)

$LeaderAgreement \triangleq$
$\land \exists\, l \in V : \forall\, v \in V : round[v] = 3 \Rightarrow leader[v] = l$
$\land \forall\, v \in V : round[v] \leq 3$

$Spec \triangleq$
$\land\ Init$
$\land \Box[Next \land LeaderAgreement']_{vars}$
$\land \forall\, v \in V,\ txs \in TxSet,\ h \in H :$
$\land \text{WF}_{vars}(StartRound(v) \land round[v] \leq 2)$
$\land \text{WF}_{vars}(GetTxSet(v,\ txs))$
$\land \text{WF}_{vars}(Vote(v))$
$\land \text{WF}_{vars}(Accept(v,\ h))$
$\land \text{WF}_{vars}(Confirm(v,\ h))$

The type-safety invariant:

$TypeOkay \triangleq$
$\land\ txSetForBalloting \in [V \to TxSet \cup \{Bot\}]$
$\land\ voted \in [V \to \text{SUBSET } H]$
$\land\ accepted \in [V \to \text{SUBSET } H]$
$\land\ round \in [V \to Nat]$
$\land\ candidates \in [V \to \text{SUBSET } TxSet]$
$\land\ preImage \in [V \to [H \to TxSet \cup \{Bot\}]]$
$\land\ leader \in [V \to V \cup \{Bot\}]$

Liveness: if a validator enters balloting, then eventually all do.

$Liveness \triangleq$
$\forall\, v \in V : \Box(txSetForBalloting[v] \neq Bot$
$\Rightarrow \exists\, t \in TxSet : \Diamond(\forall\, w \in V : txSetForBalloting[w] = t))$

Liveness: eventually, all converge on a txset for balloting.

$Liveness2 \triangleq$
$\exists\, t \in TxSet : \Diamond(\forall\, v \in V : txSetForBalloting[v] = t)$

Definition for model-checking:

Concrete hashing for the model-checker:

$TestH \triangleq 1 \,..\, Cardinality(TxSet)$

$TestHash(b) \triangleq$
LET $f \triangleq$ CHOOSE $f \in [TxSet \to H] : \forall\, txs1,\, txs2 \in TxSet : txs1 \neq txs2 \Rightarrow f[txs1] \neq f[txs2]$
IN $f[b]$

Debugging canaries:

$Canary2 \triangleq \forall\, v \in V : Cardinality(candidates[v]) \leq 1$

$Canary3 \triangleq \forall\, v \in V : txSetForBalloting[v] = Bot$

$TestQuorums \triangleq \{Q \in \text{SUBSET } V : 2 * Cardinality(Q) > Cardinality(V)\}$

$TestBlocking \triangleq \{Bl \in \text{SUBSET } V : Cardinality(Bl) > 1\}$

\ * Modification History
\ * Last modified *Fri Mar* 31 11:17:40 *PDT* 2023 by *nano*
\ * Created *Fri Jan* 13 09:09:00 *PST* 2023 by *nano*