MODULE *IscpBatchTimestamp*

Let's assume the transaction currently being built is t_i and the previous one is t_{i-1}. The following requirements apply to the timestamp $t_i.ts$ of the transaction t_i:

1. Transaction timestamps are non-decreasing function in a chain, i.e.
$$t_i.ts \geq t_{i-1}.ts.$$
2. A transaction timestamp is not smaller than the timestamps of request transactions taken as inputs in t_i, i.e.
$$\forall r \in t_i.req : t_i.ts \geq t_i.req[r].tx.ts,$$
where $t_i.req$ is a list of requests processed as inputs in the transaction t_i, $t_i.req[r]$ is a particular request and $t_i.req[r].tx$ is a transaction the request belongs to. This property is modelled bellow as the formula *Invariant*.

The initial attempt was to use the timestamp $t_i.ts$ as a median of timestamps proposed by the committee nodes (accepted to participate in the transaction t_i by the ACS procedure). This approach conflicts with the rules of selecting requests for the batch (take requests that are mentioned in at least $F+1$ proposals). In this way it is possible that the median is smaller than some request transaction timestamp .

In this document we model the case, when we take maximum of the proposed timestamps excluding F highest values. This value is close to the 66th percentile (while median is the 50th percentile). In this case all the requests selected to the batch will have timestamp lower than the batch timestamp IF THE BATCH PROPOSALS MEET THE CONDITION (modelled bellow by the formula *ProposalValid*)
$$\forall p \in batchProposals : \forall r \in p.req : p.ts \geq p.req[r].tx.ts.$$

It is possible that this rule can be violated, because of the byzantine nodes. The specification bellow shows, that property (2) can be violated, in the case of byzantine node sending timestamp lower than the requests in the proposal.

The receiving node thus needs to check, if the proposals are correct. For this check it must have all the request transactions received before deciding the final batch. The invalid batch proposals cannot be used as it. Removing them will decrease number of requests included into the final batch (because requests are included if mentioned in $F+1$ proposals). It is safe however on the receiver side to "fix" such proposals by setting their timestamp to the highest transaction timestamp of the requests in the proposal.

EXTENDS $Naturals$, $FiniteSets$, $TLAPS$, $FiniteSetTheorems$, $NaturalsInduction$

CONSTANT $Time$ — A set of timestamps, represented as natural numbers to have $\leq$.

CONSTANT $Nodes$ — A set of node identifiers.

CONSTANT $Byzantine$ — A set of byzantine node identifiers.

ASSUME $ConstantAssms \triangleq$
$$\begin{aligned}
&\land IsFiniteSet(Time) \land Time \neq \{\} \land Time \subseteq Nat \\
&\land IsFiniteSet(Nodes) \land Nodes \neq \{\} \\
&\land Byzantine \subseteq Nodes
\end{aligned}$$

$Requests \triangleq Time$ — Assume requests are identified by timestamps of their TX only.

VARIABLE $acsNodes$ — Nodes decided to be part of the round by the ACS.

VARIABLE $npRq$ — Node proposal: A set of requests.

VARIABLE $npTS$ — Node proposal: Timestamp.

$vars \triangleq \langle acsNodes, npRq, npTS \rangle$

$N \triangleq Cardinality(Nodes)$

$F \triangleq \text{CHOOSE } F \in 0\,..\,N :$
$\quad \land \; N \geq 3 * F + 1$ — *Byzantine* quorum assumption.
$\quad \land \; \forall f \in 0\,..\,N : N \geq 3 * f + 1 \Rightarrow F \geq f$ — Consider maximal possible F.
$\text{ASSUME } \mathit{ByzantineAssms} \triangleq$
$\quad \land F \in \mathit{Nat}$ — Implies CHOOSE found a suitable value.
$\quad \land N \geq 3 * F + 1$ — Standard byzantine Quorum assumption.
$\quad \land (N \geq 4 \Rightarrow F \geq 1)$ — Just to double-check in *TLC*.

$\mathit{FQuorums} \triangleq \{q \in \text{SUBSET } \mathit{Nodes} : \mathit{Cardinality}(q) = F\}$
$\mathit{F1Quorums} \triangleq \{q \in \text{SUBSET } \mathit{Nodes} : \mathit{Cardinality}(q) = F + 1\}$
$\mathit{NFQuorums} \triangleq \{q \in \text{SUBSET } \mathit{Nodes} : \mathit{Cardinality}(q) = N - F\}$

BatchRqs is a set of requests selected to the batch. Requests are selected to a batch, if they are mentioned at least in $F + 1$ proposals.

$\mathit{BatchRq}(rq) \triangleq \exists\, q \in \mathit{F1Quorums} :$
$\qquad \land\, q \subseteq \mathit{acsNodes}$
$\qquad \land\, \forall\, n \in q : rq \in \mathit{npRq}[n]$
$\mathit{BatchRqs} \triangleq \{rq \in \mathit{Requests} : \mathit{BatchRq}(rq)\}$

$\mathit{BatchTS}(ts)$ is a predicate, that is true for the timestamp that should be considered as a batch timestamp. It must be maximal of the batch proposals, excluding F greatest ones.

$\mathit{SubsetTS}(s) \triangleq \{\mathit{npTS}[n] : n \in s\}$
$\mathit{BatchTS}(ts) \triangleq$
$\quad \forall\, q \in \mathit{FQuorums} : ($
$\qquad \land\, q \subseteq \mathit{acsNodes}$
$\qquad \land\, \forall\, x \in q,\ y \in \mathit{acsNodes} \setminus q : \mathit{npTS}[x] \geq \mathit{npTS}[y]$
$\quad) \Rightarrow ($
$\qquad \land\;\; ts \in \mathit{SubsetTS}(\mathit{acsNodes} \setminus q)$
$\qquad \land\;\; \forall\, x \in \mathit{SubsetTS}(\mathit{acsNodes} \setminus q) : ts \geq x$
$\qquad \land\;\; \forall\, x \in \mathit{SubsetTS}(q) : ts \leq x$
$\quad)$

A batch proposal is valid, if its timestamp is not less than timestamps of all the request transactions included to the proposal.

$\mathit{ProposalValid}(n) \triangleq \forall\, rq \in \mathit{npRq}[n] : rq \leq \mathit{npTS}[n]$

$\mathit{Init} \triangleq$
$\quad \land\, \mathit{acsNodes} \in \text{SUBSET } \mathit{Nodes} \land \mathit{Cardinality}(\mathit{acsNodes}) \geq N - F$
$\quad \land\, \mathit{npRq} \in [\mathit{acsNodes} \rightarrow (\text{SUBSET } \mathit{Requests}) \setminus \{\{\}\}]$
$\quad \land\, \mathit{npTS} \in [\mathit{acsNodes} \rightarrow \mathit{Time}]$
$\quad \land\, \forall\, n \in (\mathit{acsNodes} \setminus \mathit{Byzantine}) : \mathit{ProposalValid}(n)$ — Fair node proposals are valid.
$\mathit{Next} \triangleq \text{UNCHANGED } \mathit{vars}$ — Only for model checking in *TLC*.
$\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}}$

$\mathit{TypeOK} \triangleq$
$\quad \land\, \mathit{acsNodes} \subseteq \mathit{Nodes}$

```
  ∧ npRq ∈ [acsNodes → SUBSET Requests]
  ∧ npTS ∈ [acsNodes → Time]

Invariant ≜
  ∀ ts ∈ Time, rq ∈ BatchRqs : BatchTS(ts) ⇒ rq ≤ ts

THEOREM Spec ⇒ □TypeOK ∧ □Invariant
  PROOF OMITTED   Checked with TLC, and check the proofs bellow.
--------------------------------------------------------------------------------
LEMMA SubsetsAllCardinalities ≜
  ASSUME NEW S, IsFiniteSet(S)
  PROVE ∀ x ∈ 0 . . Cardinality(S) : ∃ q ∈ SUBSET S : Cardinality(q) = x
PROOF
⟨1⟩ DEFINE P(x) ≜ x ≤ Cardinality(S) ⇒ ∃ q ∈ SUBSET S : Cardinality(q) = x
⟨1⟩1. ∀ x ∈ Nat : P(x)
  ⟨2⟩1. P(0)BY FS_EmptySet
  ⟨2⟩2. ∀ x ∈ Nat : P(x) ⇒ P(x + 1)
    ⟨3⟩1. TAKE x ∈ Nat
    ⟨3⟩2. HAVE P(x)
    ⟨3⟩3. HAVE x + 1 ≤ Cardinality(S)
    ⟨3⟩4. PICK qx ∈ SUBSET S : Cardinality(qx) = x
          BY ⟨3⟩2, ⟨3⟩3, FS_CardinalityType
    ⟨3⟩5. PICK x1 ∈ S : x1 ∉ qx
          BY ⟨3⟩3, ⟨3⟩4
    ⟨3⟩6. WITNESS qx ∪ {x1} ∈ SUBSET S
    ⟨3⟩7. Cardinality(qx ∪ {x1}) = x + 1
          BY ⟨3⟩4, ⟨3⟩5, FS_AddElement, FS_Subset
    ⟨3⟩ QED BY ⟨3⟩7
  ⟨2⟩3. QED BY ⟨2⟩1, ⟨2⟩2, NatInduction
⟨1⟩2. QED BY ⟨1⟩1

LEMMA NatSubsetHasMax ≜
  ASSUME NEW S, IsFiniteSet(S), S ≠ {}, S ∈ SUBSET Nat
  PROVE ∃ n ∈ S : ∀ s ∈ S : s ≤ n
⟨1⟩ DEFINE P(x) ≜ x ≠ {} ∧ x ⊆ S ⇒ ∃ n ∈ x : ∀ s ∈ x : s ≤ n
⟨1⟩ SUFFICES ASSUME TRUEPROVE P(S)OBVIOUS
⟨1⟩0. IsFiniteSet(S)OBVIOUS
⟨1⟩1. P({})OBVIOUS
⟨1⟩2. ASSUME NEW T, NEW x, IsFiniteSet(T), P(T), x ∉ TPROVE P(T ∪ {x})
  ⟨2⟩1.CASE ∀ t ∈ T : x ≥ t
    ⟨3⟩0. HAVE T ∪ {x} ≠ {} ∧ T ∪ {x} ⊆ S
    ⟨3⟩1. WITNESS x ∈ T ∪ {x}
    ⟨3⟩ QED BY ⟨2⟩1, ⟨3⟩0
  ⟨2⟩2.CASE ¬∀ t ∈ T : x ≥ t
    ⟨3⟩4.CASE T = {} ∨ ¬T ⊆ SBY ⟨3⟩4
    ⟨3⟩5.CASE T ≠ {} ∧ T ⊆ S
```

```
        ⟨4⟩1. P(T) BY ⟨1⟩2
        ⟨4⟩2. ∃ n ∈ T : ∀ s ∈ T : s ≤ n BY ⟨4⟩1, ⟨3⟩5
        ⟨4⟩ QED BY ⟨4⟩2, ⟨3⟩5, ⟨2⟩2
      ⟨3⟩ QED BY ⟨3⟩4, ⟨3⟩5
    ⟨2⟩3. QED BY ⟨2⟩1, ⟨2⟩2
⟨1⟩ HIDE DEF P
⟨1⟩ QED BY ONLY ⟨1⟩0, ⟨1⟩1, ⟨1⟩2, FS_Induction

THEOREM SpecTypeOK ≜ Spec ⇒ □TypeOK
  ⟨1⟩1. Init ⇒ TypeOK BY DEF Init, TypeOK
  ⟨1⟩2. TypeOK ∧ [Next]_vars ⇒ TypeOK' BY DEF vars, TypeOK, Next
  ⟨1⟩3. QED BY ⟨1⟩1, ⟨1⟩2, PTL DEF Spec

THEOREM SpecInvariant ≜ Byzantine = {} ∧ Spec ⇒ □Invariant
  ⟨1⟩ SUFFICES ASSUME Byzantine = {} PROVE Spec ⇒ □Invariant OBVIOUS
  ⟨1⟩1. TypeOK ∧ Init ⇒ Invariant
    ⟨2⟩ SUFFICES ASSUME TypeOK, Init PROVE Invariant OBVIOUS
    ⟨2⟩ USE DEF Invariant
    ⟨2⟩ TAKE ts ∈ Time, rq ∈ BatchRqs
    ⟨2⟩ HAVE BatchTS(ts) PROVE : rq ≤ ts
    ⟨2⟩1. ∀ q1 ∈ F1Quorums, q2 ∈ NFQuorums : q1 ∩ q2 ≠ {}
      ⟨3⟩ TAKE q1 ∈ F1Quorums, q2 ∈ NFQuorums
      ⟨3⟩1. N ∈ Nat ∧ F ∈ Nat BY ONLY ConstantAssms, ByzantineAssms, FS_CardinalityType DEF N, F
      ⟨3⟩2. Cardinality(q1) + Cardinality(q2) > Cardinality(Nodes) BY ONLY ⟨3⟩1 DEF N, F1Quorums, NFQ
      ⟨3⟩3. q1 ⊆ Nodes ∧ q2 ⊆ Nodes BY ONLY DEF F1Quorums, NFQuorums
      ⟨3⟩4. QED BY ONLY ⟨3⟩2, ⟨3⟩3, FS_MajoritiesIntersect, ConstantAssms
    ⟨2⟩2. ∀ rr ∈ BatchRqs : ∃ q ∈ F1Quorums : ∀ n ∈ q : rr ∈ npRq[n] BY DEF BatchRqs, BatchRq
    ⟨2⟩3. ∀ nn ∈ acsNodes : ProposalValid(nn) BY DEF Init
    ⟨2⟩4. acsNodes ⊆ Nodes BY DEF Init
    ⟨2⟩5. Cardinality(acsNodes) − F > 0
      ⟨3⟩1. Cardinality(acsNodes) ∈ Nat BY ⟨2⟩4, FS_CardinalityType, FS_Subset, ConstantAssms
      ⟨3⟩2. F ∈ Nat BY ByzantineAssms
      ⟨3⟩3. N ∈ Nat BY ConstantAssms, FS_CardinalityType DEF N
      ⟨3⟩4. Cardinality(acsNodes) ≥ N − F BY DEF Init
      ⟨3⟩5. N − F ≥ 2 ∗ F + 1 BY ByzantineAssms, ⟨3⟩2, ⟨3⟩3
      ⟨3⟩6. Cardinality(acsNodes) > F BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4, ⟨3⟩5, ByzantineAssms
      ⟨3⟩ QED BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩6
    ⟨2⟩6. Cardinality(acsNodes) − F ≥ 0 BY ⟨2⟩5
    ⟨2⟩7. ∀ fq ∈ FQuorums, f1q ∈ F1Quorums : ¬f1q ⊆ fq
      ⟨3⟩1. TAKE fq ∈ FQuorums, f1q ∈ F1Quorums
      ⟨3⟩2. SUFFICES ASSUME f1q ⊆ fq PROVE FALSE OBVIOUS
      ⟨3⟩3. IsFiniteSet(f1q) ∧ IsFiniteSet(fq) BY ConstantAssms, FS_Subset DEF FQuorums, F1Quorums
      ⟨3⟩4. Cardinality(f1q) ≤ Cardinality(fq) BY ⟨3⟩2, ⟨3⟩3, FS_Subset
      ⟨3⟩5. Cardinality(f1q) > Cardinality(fq) BY ByzantineAssms DEF F1Quorums, FQuorums
      ⟨3⟩q. QED BY ⟨3⟩3, ⟨3⟩4, ⟨3⟩5, FS_CardinalityType
```

$\langle 2\rangle 8.\ F \in Nat \land F \geq 0 \land F \leq N \land F + 1 \leq N$
$\langle 3\rangle 1.\ F \in Nat$BY $ByzantineAssms$
$\langle 3\rangle 2.\ F \geq 0$BY $\langle 3\rangle 1$, $ConstantAssms$ DEF F
$\langle 3\rangle 3.\ N \in Nat$BY $ConstantAssms$, $FS\_CardinalityType$ DEF N
$\langle 3\rangle 4.\ F \leq N$BY ONLY $\langle 3\rangle 1$, $\langle 3\rangle 3$, $ConstantAssms$, $ByzantineAssms$ DEF F
$\langle 3\rangle 5.\ F + 1 \leq N$BY ONLY $\langle 3\rangle 1$, $\langle 3\rangle 3$, $ConstantAssms$, $ByzantineAssms$ DEF F
$\langle 3\rangle$q. QED BY ONLY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 4$, $\langle 3\rangle 5$
$\langle 2\rangle 9.\ FQuorums \neq \{\} \land F1Quorums \neq \{\} \land NFQuorums \neq \{\}$
BY $\langle 2\rangle 8$, $FS\_CardinalityType$, $ConstantAssms$, $SubsetsAllCardinalities$
DEF $FQuorums$, $F1Quorums$, $NFQuorums$, N
$\langle 2\rangle 10.$ PICK $fq \in FQuorums : fq \subseteq acsNodes \land \forall x \in fq,\ y \in acsNodes \setminus fq : npTS[x] \geq npTS[y]$
$\langle 3\rangle 1.$ SUFFICES $\exists fq \in FQuorums : fq \subseteq acsNodes \land \forall x \in fq,\ y \in acsNodes \setminus fq : npTS[x] \geq npTS[y]$OBV
$\langle 3\rangle 2.\ Cardinality(acsNodes) \geq N - F$BY DEF $Init$
$\langle 3\rangle 3.\ N - F \geq F$BY $\langle 2\rangle 8$, $ByzantineAssms$, $ConstantAssms$, $FS\_CardinalityType$ DEF N
$\langle 3\rangle 4.\ N - F > 0$BY $\langle 2\rangle 8$, $ByzantineAssms$, $ConstantAssms$, $FS\_CardinalityType$ DEF N
$\langle 3\rangle 5.\ N \in Nat$BY $FS\_CardinalityType$, $ConstantAssms$ DEF N
$\langle 3\rangle 6.\ acsNodes \subseteq Nodes$BY DEF $Init$
$\langle 3\rangle 7.\ acsNodes \neq \{\}$BY ONLY $\langle 3\rangle 2$, $\langle 3\rangle 4$, $\langle 3\rangle 5$, $\langle 3\rangle 6$, $\langle 2\rangle 8$, $FS\_EmptySet$ DEF $Init$
$\langle 3\rangle 8.\ IsFiniteSet(acsNodes)$BY FS_Subset, $ConstantAssms$ DEF $Init$
$\langle 3\rangle 9.$ PICK $card \in Nat : card = Cardinality(acsNodes)$BY $\langle 3\rangle 8$, $FS\_CardinalityType$
$\langle 3\rangle 10.\ card \geq 0 \land card \geq N - F \land card \geq F$BY $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 2\rangle 8$, $\langle 3\rangle 5$, $\langle 3\rangle 9$
$\langle 3\rangle 11.$ PICK $q \in$ SUBSET $acsNodes : Cardinality(q) = F \land \forall x \in q,\ y \in acsNodes \setminus q : npTS[x] \geq npTS[y$
$\langle 4\rangle\ \forall q \in$ SUBSET $acsNodes : acsNodes \setminus q \subseteq Nodes$BY DEF $Init$
$\langle 4\rangle\ \forall q \in$ SUBSET $acsNodes : acsNodes \setminus q \subseteq acsNodes$BY DEF $Init$
$\langle 4\rangle\ \forall n \in acsNodes : npTS[n] \in Nat$BY $ConstantAssms$ DEF $TypeOK$
$\langle 4\rangle\ \forall c \in 0 \,..\, card : \exists q \in$ SUBSET $acsNodes : Cardinality(q) = c \land \forall x \in q,\ y \in acsNodes \setminus q : npTS[x]$
$\langle 5\rangle$ DEFINE $P(c) \triangleq c \leq card \Rightarrow \exists q \in$ SUBSET $acsNodes : Cardinality(q) = c \land \forall x \in q,\ y \in acsNode$
$\langle 5\rangle 1.$ SUFFICES ASSUME TRUEPROVE $\forall c \in Nat : P(c)$OBVIOUS
$\langle 5\rangle 2.\ P(0)$BY $\langle 3\rangle 9$, $FS\_EmptySet$
$\langle 5\rangle 3.\ \forall c \in Nat : P(c) \Rightarrow P(c + 1)$
$\langle 6\rangle 1.$ TAKE $c \in Nat$
$\langle 6\rangle 2.$ HAVE $P(c)$
$\langle 6\rangle 3.$ HAVE $c + 1 \leq card$
$\langle 6\rangle 4.$ PICK $q \in$ SUBSET $acsNodes : Cardinality(q) = c \land (\forall x \in q,\ y \in acsNodes \setminus q : npTS[x] \geq np$
$\langle 6\rangle 5.$ PICK $x \in (acsNodes \setminus q) : \forall xx \in acsNodes \setminus q : npTS[x] \geq npTS[xx]$
$\langle 7\rangle 1.\ Cardinality(acsNodes) \geq c + 1$BY $\langle 6\rangle 3$, $\langle 3\rangle 9$
$\langle 7\rangle 2.\ Cardinality(q) = c$BY $\langle 6\rangle 4$
$\langle 7\rangle$ DEFINE $Q \triangleq acsNodes \setminus q$
$\langle 7\rangle 3.\ Q \neq \{\}$BY $\langle 7\rangle 1$, $\langle 7\rangle 2$, FS_Subset
$\langle 7\rangle 4.\ IsFiniteSet(Q)$BY $\langle 3\rangle 8$, FS_Subset
$\langle 7\rangle 5.\ Q \in$ SUBSET $acsNodes$BY DEF $TypeOK$
$\langle 7\rangle 6.$ PICK $tt \in \{npTS[xx] : xx \in Q\} : \forall ttt \in \{npTS[xx] : xx \in Q\} : ttt \leq tt$
$\langle 8\rangle$ DEFINE $QTS \triangleq \{npTS[xx] : xx \in Q\}$
$\langle 8\rangle$ HIDE DEF Q
$\langle 8\rangle 1.\ npTS \in [acsNodes \rightarrow Time]$BY DEF $TypeOK$

$\langle 8\rangle 2.$ $QTS \neq \{\}$BY ONLY $\langle 7\rangle 3$, $\langle 7\rangle 5$, $\langle 8\rangle 1$
$\langle 8\rangle 3.$ $QTS \in$ SUBSET NatBY DEF $TypeOK$, Q
$\langle 8\rangle 4.$ $IsFiniteSet(QTS)$BY ONLY $\langle 7\rangle 4$, FS_Image
$\langle 8\rangle 5.$ $\exists\, tt \in QTS : \forall\, x \in QTS : tt \geq x$BY ONLY $\langle 8\rangle 2$, $\langle 8\rangle 3$, $\langle 8\rangle 4$, $NatSubsetHasMax$
$\langle 8\rangle 6.$ PICK $tt \in QTS : \forall\, x \in QTS : tt \geq x$BY $\langle 8\rangle 5$
$\langle 8\rangle 7.$ WITNESS $tt \in QTS$
$\langle 8\rangle 8.$ QED BY $\langle 8\rangle 6$
$\langle 7\rangle 7.$ $\exists\, nn \in Q : npTS[nn] = tt$BY ONLY $\langle 7\rangle 6$, $\langle 7\rangle 3$, $TypeOK$ DEF $TypeOK$
$\langle 7\rangle 8.$ PICK $nn \in Q : \ npTS[nn] = tt$BY $\langle 7\rangle 7$
$\langle 7\rangle 9.$ WITNESS $nn \in Q$
$\langle 7\rangle$ QED BY $\langle 7\rangle 6$, $\langle 7\rangle 8$
$\langle 6\rangle 6.$ $q \cup \{x\} \in$ SUBSET $acsNodes$BY $\langle 6\rangle 4$, $\langle 6\rangle 5$
$\langle 6\rangle 7.$ WITNESS $q \cup \{x\} \in$ SUBSET $acsNodes$
$\langle 6\rangle 8.$ $IsFiniteSet(q)$BY $\langle 3\rangle 8$, $\langle 6\rangle 4$, FS_Subset
$\langle 6\rangle 9.$ $Cardinality(q \cup \{x\}) = c + 1$BY $FS\_AddElement$, $\langle 6\rangle 5$, $\langle 6\rangle 4$, $\langle 6\rangle 8$
$\langle 6\rangle 10.$ $\forall\, xx \in q \cup \{x\},\ y \in acsNodes \setminus (q \cup \{x\}) : npTS[xx] \geq npTS[y]$
$\langle 7\rangle 1.$ TAKE $xx \in q \cup \{x\},\ y \in acsNodes \setminus (q \cup \{x\})$
$\langle 7\rangle 2.$CASE $xx \ = x$BY $\langle 7\rangle 2$, $\langle 6\rangle 5$
$\langle 7\rangle 3.$CASE $xx \ \in q$BY $\langle 7\rangle 3$, $\langle 6\rangle 4$
$\langle 7\rangle 4.$ QED BY $\langle 7\rangle 2$, $\langle 7\rangle 3$
$\langle 6\rangle 11.$ QED BY $\langle 6\rangle 9$, $\langle 6\rangle 10$
$\langle 5\rangle 4.$ HIDE DEF P
$\langle 5\rangle 5.$ QED BY $\langle 5\rangle 2$, $\langle 5\rangle 3$, $NatInduction$
$\langle 4\rangle$ QED BY $\langle 3\rangle 8$, $\langle 3\rangle 9$, $\langle 3\rangle 10$, $\langle 2\rangle 8$, FS_Subset, $FS\_CardinalityType$, $SubsetsAllCardinalities$
$\langle 3\rangle 12.$ $q \in FQuorums \wedge \forall\, x \in q,\ y \in acsNodes \setminus q : npTS[x] \geq npTS[y]$BY $\langle 3\rangle 11$, $\langle 3\rangle 6$ DEF $FQuorums$
$\langle 3\rangle 13.$ $q \in FQuorums$BY $\langle 3\rangle 11$, $\langle 3\rangle 6$ DEF $FQuorums$
$\langle 3\rangle 14.$ WITNESS $q \in FQuorums$
$\langle 3\rangle 15.$ QED BY $\langle 3\rangle 12$, $\langle 3\rangle 14$
$\langle 2\rangle 11.$ $\forall\, x \in BatchRqs : x \leq ts$
$\langle 3\rangle 1.$ TAKE $x \in BatchRqs$
$\langle 3\rangle 2.$ $x \in Requests \wedge BatchRq(x)$BY $\langle 3\rangle 1$ DEF $BatchRqs$
$\langle 3\rangle 3.$ PICK $xf1q \in F1Quorums : xf1q \subseteq acsNodes \wedge \forall\, n \in xf1q : x \in npRq[n]$BY $\langle 3\rangle 2$ DEF $BatchRq$
$\langle 3\rangle 4.$ $xf1q \setminus fq \quad \neq \{\}$
$\langle 4\rangle 1.$ $Cardinality(xf1q) = F + 1$BY $\langle 3\rangle 3$ DEF $F1Quorums$
$\langle 4\rangle 2.$ $Cardinality(fq) = F$BY $\langle 2\rangle 10$ DEF $FQuorums$
$\langle 4\rangle 3.$ $F \in Nat$BY $ByzantineAssms$
$\langle 4\rangle 4.$ $xf1q \subseteq Nodes \wedge fq \subseteq Nodes$BY $\langle 3\rangle 3$, $\langle 2\rangle 10$ DEF $F1Quorums$, $FQuorums$
$\langle 4\rangle 5.$ $IsFiniteSet(xf1q) \wedge IsFiniteSet(fq)$BY $\langle 4\rangle 4$, $ConstantAssms$, FS_Subset
$\langle 4\rangle 6.$ QED BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 5$, FS_Subset
$\langle 3\rangle 5.$ $\forall\, n \in (xf1q \setminus fq) : \forall\, r \in npRq[n] : r \leq ts$
$\langle 4\rangle 1.$ $xf1q \setminus fq \subseteq acsNodes$BY $\langle 2\rangle 10$, $\langle 3\rangle 3$
$\langle 4\rangle 2.$ TAKE $xn \in (xf1q \setminus fq)$
$\langle 4\rangle 3.$ TAKE $xr \in npRq[xn]$
$\langle 4\rangle 4.$ $xr \in Nat$BY $\langle 4\rangle 3$, $\langle 4\rangle 1$, $ConstantAssms$ DEF $TypeOK$, $Requests$
$\langle 4\rangle 5.$ $ts \in Nat$BY $ConstantAssms$

```
        ⟨4⟩6. npTS[xn] ∈ NatBY ⟨4⟩2, ⟨4⟩1, ConstantAssms DEF TypeOK
        ⟨4⟩7. npTS[xn] ≤ ts
          ⟨5⟩1. xn ∈ acsNodesBY ⟨4⟩2, ⟨4⟩1
          ⟨5⟩2. xn ∉ fqBY ⟨4⟩2
          ⟨5⟩3. ∧ ts ∈ SubsetTS(acsNodes \ fq)
                ∧ ∀ xx ∈ SubsetTS(acsNodes \ fq) : ts ≥ xx
                ∧ ∀ xx ∈ SubsetTS(fq) : ts ≤ xx
                BY ⟨2⟩10  DEF BatchTS
          ⟨5⟩4. QED BY ⟨5⟩1, ⟨5⟩2, ⟨5⟩3  DEF SubsetTS
        ⟨4⟩8. xr ≤ npTS[xn]
          ⟨5⟩ ProposalValid(xn)BY ⟨4⟩1  DEF Init
          ⟨5⟩ QED BY  DEF ProposalValid
        ⟨4⟩9. QED BY ONLY ⟨4⟩7, ⟨4⟩8, ⟨4⟩4, ⟨4⟩5, ⟨4⟩6
      ⟨3⟩6. ∃ n ∈ (xf1q \ fq) : x ∈ npRq[n]BY ⟨3⟩4, ⟨3⟩3
      ⟨3⟩7. QED BY ⟨3⟩5, ⟨3⟩6
    ⟨2⟩12. QED BY ⟨2⟩11
  ⟨1⟩2. Invariant ∧ [Next]_vars ⇒ Invariant'
    ⟨2⟩1. SUFFICES ASSUME InvariantPROVE [Next]_vars ⇒ Invariant'
          OBVIOUS
    ⟨2⟩2. UNCHANGED vars ⇒ (Invariant')
          BY ⟨2⟩1  DEF vars, Invariant, BatchRq, BatchRqs, BatchTS,
                       ProposalValid, SubsetTS
    ⟨2⟩3. SUFFICES ASSUME NextPROVE Invariant'
          BY ⟨2⟩2
    ⟨2⟩4. QED BY ⟨2⟩1, ⟨2⟩3  DEF vars, Next, Invariant, BatchRq,
                  BatchRqs, BatchTS, ProposalValid, SubsetTS
  ⟨1⟩q. QED BY ⟨1⟩1, ⟨1⟩2, PTL, SpecTypeOK DEF Spec, vars
```

Counter-example with *Nodes* = 101 .. 104, *Byzantine* = {104}, *Time* = 1 .. 3:

PropposedRq: (101 :> {1} @@ 102 :> {1} @@ 103 :> {2} @@ 104 :> {2}),

PropposedTS: (101 :> 1 @@ 102 :> 1 @@ 103 :> 2 @@ 104 :> 1),

BatchRq: {1, 2},

BatchTS: 1