Stellar Quorum Systems

Giuliano Losa
Galois, Inc., USA
giuliano@galois.com

August 3, 2019

Abstract

We formalize the static properties of personal Byzantine quorum systems (PBQSSs) and Stellar quorum systems, as described in the paper “Stellar Consensus by Reduction”, to appear at DISC 2019.

Contents

1 Personal Byzantine quorum systems 2
  1.1 The set of participants not blocked by malicious participants 3
  1.2 Consensus clusters and intact sets 3

2 Stellar quorum systems 5
  2.1 Properties of blocking sets 6

3 Reachability through a set 9

4 Elementary quorums 10
This theory formalizes some of the results appearing in the paper "Stellar Consensus By Reduction"[1]. We prove static properties of personal Byzantine quorum systems and Stellar quorum systems.

theory Stellar-Quorums
  imports Main
begin

1 Personal Byzantine quorum systems

locale personal-quorums
  fixes quorum-of :: 'node ⇒ 'node set ⇒ bool
  assumes quorum-assm:\ p p' . [p ∈ W; quorum-of p Q; p' ∈ Q∩W] ⇒ quorum-of p' Q
  — In other words, a quorum (of some participant) is a quorum of all its members.

begin

definition blocks where
  — Set \( R \) blocks participant \( p \).
  blocks R p ≡ ∀ Q . quorum-of p Q −→ Q ∩ R ≠ {}

abbreviation blocked-by where blocked-by R ≡ \{ p . blocks R p \}

lemma blocked-blocked-subset-blocked:
  blocked-by (blocked-by R) ⊆ blocked-by R
proof −
  have False if p ∈ blocked-by (blocked-by R) and p ∉ blocked-by R for p
  proof −
    have Q ∩ blocked-by R ≠ {} if quorum-of p Q for Q
      using (p ∈ blocked-by (blocked-by R)) by unfolding blocks-def by auto
    have Q ∩ R ≠ {} if quorum-of p Q for Q
      proof −
        obtain p' where p' ∈ blocked-by R and p' ∈ Q
          by (meson Int-emptyI (\{ Q . quorum-of p Q ⇒ Q ∩ blocked-by R ≠ {}\})
          ⟨quorum-of p Q⟩)
        hence quorum-of p' Q
          using quorum-assm that by blast
        with p' ∈ blocked-by R; show Q ∩ R ≠ {}
          using blocks-def by auto
      qed
      hence p ∈ blocked-by R by (simp add: blocks-def)
      thus False using that(2) by auto
  qed
  thus blocked-by (blocked-by R) ⊆ blocked-by R
  by blast
  qed

end
We now add the set of correct participants to the model.

```plaintext
locale with-w = personal-quorums quorum-of for quorum-of :: 'node ⇒ 'node set ⇒ bool +
fixes W::'node set — W is the set of correct participants
begin

abbreviation B where B ≡ − W
— B is the set of malicious participants.

definition quorum-of-set where quorum-of-set S Q ≡ ∃ p ∈ S . quorum-of p Q

1.1 The set of participants not blocked by malicious participants

definition L where L ≡ W − (blocked-by B)

lemma l2: p ∈ L ⇒ ∃ Q ⊆ W . quorum-of p Q
  unfolding L-def blocks-def using DiffD2 by auto

lemma l3: — If a participant is not blocked by the malicious participants, then it has a quorum consisting exclusively of correct participants which are not blocked by the malicious participants.
  assumes p ∈ L shows ∃ Q ⊆ L . quorum-of p Q
  proof —
    have False if \( \bigwedge Q . \text{quorum-of p Q} \implies Q \cap (-L) \neq \{\} \)
    proof —
      obtain Q where quorum-of p Q and Q ⊆ W
      using l2 {p ∈ L by auto
      have Q ∩ (-L) ≠ \{\} using that 'quorum-of p Q' by simp
      obtain p' where p' ∈ Q ∩ (-L) and quorum-of p' Q
      using ⟨Q ∩ -L ≠ \{\}, 'quorum-of p Q'⟩ inf.left-idem quorum-assm by fastforce
      hence Q ∩ B ≠ \{\} unfolding L-def
      using CollectD Compl-Diff-eq Int-iff inf-le1 personal-quorums,blocks-def personal-quorums-axioms
      subset-empty by fastforce
      thus False using ⟨Q ⊆ W⟩ by auto
      qed
      thus ?thesis by (metis disjoint-eq-subset-Compl double-complement)
    qed

1.2 Consensus clusters and intact sets

definition is-intertwined where
— This definition is not used in this theory, but we include it to formalize the notion of intertwined set appearing in the DISC paper.
  is-intertwined S ≡ S ⊆ W
    ∧ (∀ Q Q' . quorum-of-set S Q ∧ quorum-of-set S Q' → W ∩ Q ∩ Q' ≠ \{\})

definition is-cons-cluster where
```
— Consensus clusters
\[\text{is-cons-cluster } C \equiv C \subseteq W \land (\forall p \in C . \exists Q \subseteq C . \text{quorum-of-} p \ Q)\] 
\[\land (\forall Q Q'. \text{quorum-of-set } C Q \land \text{quorum-of-set } C Q' \rightarrow W \cap Q \cap Q' \neq \{\})\]

**definition** stellar-intact where

— This is equivalent to the notion of intact set presented in the Stellar Whitepaper [2]
\[\text{stellar-intact } I \equiv I \subseteq W \land (\forall p \in I . \exists Q \subseteq I . \text{quorum-of-} p \ Q)\] 
\[\land (\forall Q Q'. \text{quorum-of-set } I Q \land \text{quorum-of-set } I Q' \rightarrow I \cap Q \cap Q' \neq \{\})\]

**lemma** stellar-intact-imp-cons-cluster:

— Every intact set is a consensus cluster

**shows** stellar-intact I \[\implies\] is-cons-cluster I

**unfolding** stellar-intact-def is-cons-cluster-def

by blast

**lemma** cons-cluster-not-intact:

— Some consensus clusters are not intact and have no intact superset.

**shows** is-cons-cluster C \[\land\] (\forall J . C \subseteq J \rightarrow \neg stellar-intact J) nitpick[falsify=false, card 'node'=3, expect=genuine]

oops

Next we show that the union of two consensus clusters that intersect is a consensus cluster.

**theorem** cluster-union:

**assumes** is-cons-cluster C 1 and is-cons-cluster C 2 and C 1 \cap C 2 \neq \{}

**shows** is-cons-cluster (C 1 \cup C 2)

**proof** —

**have** C 1 \cup C 2 \subseteq W

using assms(1) assms(2) is-cons-cluster-def by auto

moreover

**have** \(\forall p \in (C_1 \cup C_2) . \exists Q \subseteq (C_1 \cup C_2) . \text{quorum-of-} p \ Q\)

using (is-cons-cluster C 1) (is-cons-cluster C 2) unfolding is-cons-cluster-def

by (meson UnE le-supI1 le-supI2)

moreover

**have** \(W \cap Q_1 \cap Q_2 \neq \{\})\)

if \(\text{quorum-of-set } (C_1 \cup C_2) Q_1 \and \text{quorum-of-set } (C_1 \cup C_2) Q_2\)

for \(Q_1 Q_2\)

proof —

**have** \(W \cap Q_1 \cap Q_2 \neq \{\})\) if \(\text{quorum-of-set } C Q_1 \and \text{quorum-of-set } C Q_2 \and\)

\(C = C_1 \setminus C = C_2 \text{ for } C\)

using (is-cons-cluster C 1) (is-cons-cluster C 2) (quorum-of-set (C 1 \cup C 2) Q 1)

(quorum-of-set (C 1 \cup C 2) Q 2) that

unfolding quorum-of-set-def is-cons-cluster-def by metis

moreover

**have** \(\{W \cap Q_1 \cap Q_2 \neq \{\})\) if is-cons-cluster C 1 and is-cons-cluster C 2

and C 1 \cap C 2 \neq \{}

and quorum-of-set C 1 Q 1 and quorum-of-set C 2 Q 2

for C 1 C 2 — We generalize to avoid repeating the argument twice

4
proof
  obtain p Q where quorum-of p Q and p ∈ C1 ∩ C2 and Q ⊆ C2
  using ⟨C1 ∩ C2 ≠ {⟩ (is-cons-cluster C2) unfolding is-cons-cluster-def
by blast
  have Q ∩ Q1 ≠ { using ⟨is-cons-cluster C1⟩ (quorum-of-set C1 Q1)
  unfolding is-cons-cluster-def quorum-of-set-def
  by (metis Int-assoc Int-iff inf-bot-right)
  hence quorum-of-set C2 Q1 using ⟨Q ⊆ C2⟩ (quorum-of-set C1 Q1)
quorum-assm unfolding quorum-of-set-def by blast
  thus W ∩ Q1 ∩ Q2 ≠ {} using (is-cons-cluster C2) (quorum-of-set C2 Q2)
  unfolding is-cons-cluster-def by blast
  qed
ultimately show ?thesis using assms that unfolding quorum-of-set-def by
auto
  qed
ultimately show ?thesis using assms
  unfolding is-cons-cluster-def by simp
  qed
end

2 Stellar quorum systems

locale stellar =
  fixes slices :: 'node ⇒ 'node set set — the quorum slices
  and W :: 'node set — the correct participants
  assumes slices-ne:∀ p. p ∈ W ⇒ slices p ≠ {}
begin

definition quorum where
quorum Q = ∀ p ∈ Q ∩ W . (∃ Sl ∈ slices p . Sl ⊆ Q)

definition quorum-of where quorum-of p Q = quorum Q ∧ (∃ Sl ∈ slices p . Sl ⊆ Q))
  — TODO: p ∉ W needed?

lemma quorum-union; quorum Q ⇒ quorum Q′ ⇒ quorum (Q ∪ Q′)
  unfolding quorum-def
by (metis IntE Int-iff UnE inf-sup-aci(1) sup.coboundedI1 sup.coboundedI2)

lemma I1:
  assumes p . p ∈ S ⇒ ∃ Q ⊆ S . quorum-of p Q and p ∈ S
  shows quorum-of p S using assms unfolding quorum-of-def quorum-def
  by (meson Int-iff subset-trans)

lemma is-pbqs:
  assumes quorum-of p Q and p ∈ Q
  shows quorum-of p′ Q
— This is the property required of a PBQS.

using assms
by (simp add: quorum-def quorum-of-def)

interpretation with-w quorum-of
— Stellar quorums form a personal quorum system.

unfolding with-w-def personal-quorums-def
quorum-def quorum-of-def by simp

lemma quorum-is-quorum-of-some-slice:
assumes quorum-of p Q and p ∈ W
obtains S where S ∈ slices p and S ⊆ Q
and \( \bigwedge p'. p' ∈ S \cap W \Rightarrow \text{quorum-of} p' Q \)
using assms unfolding quorum-def quorum-of-def by fastforce

lemma is-cons-cluster C =⇒ quorum C
— Every consensus cluster is a quorum.

unfolding is-cons-cluster-def
by (metis inf.order-iff l1 quorum-of-def stellar.quorum-def stellar-axioms)

2.1 Properties of blocking sets

inductive blocking-min where
— This is the set of correct participants that are eventually blocked by a set \( R \) when Byzantine processors do not take steps.

\[
[p \in W; \forall Sl \in \text{slices } p . \exists q \in Sl \cap W . q \in R \lor \text{blocking-min} R q] \Rightarrow \text{blocking-min} R p
\]

inductive-cases blocking-min-elim:blocking-min R p

inductive blocking-max where
— This is the set of participants that are eventually blocked by a set \( R \) when Byzantine processors help epidemic propagation.

\[
[p \in W; \forall Sl \in \text{slices } p . \exists q \in Sl . q \in R \cup B \lor \text{blocking-max} R q] \Rightarrow \text{blocking-max} R p
\]

inductive-cases blocking-max R p

Next we show that if \( R \) blocks \( p \) and \( p \) belongs to a consensus cluster \( S \), then \( R \cap S \neq \{\} \).

We first prove two auxiliary lemmas:

lemma cons-cluster-wb:p ∈ C =⇒ is-cons-cluster C =⇒ p ∈ W
using is-cons-cluster-def by fastforce

lemma cons-cluster-has-ne-slices:
assumes is-cons-cluster C and p ∈ C
and Sl ∈ slices p
shows Sl ≠ \{\}
using assms unfolding is-cons-cluster-def quorum-of-set-def quorum-of-def quorum-def
by (metis empty-iff inf.bot-left inf.bot-right subset_refl)
lemma cons-cluster-has-cons-cluster-slice:
  assumes is-cons-cluster C and p∈C
  obtains Sl where Sl ∈ slices p and Sl ⊆ C
  using assms unfolding is-cons-cluster-def quorum-of-set-def quorum-of-def quorum-def
  by (metis Int-commute empty-iff inf.order-iff inf-bot-right le-infI1)

theorem blocking-max-intersects-intact:
  — if R blocks p when malicious participants help epidemic propagation, and p
  belongs to a consensus cluster C, then R ∩ C ≠ {}.
  assumes blocking-max R p and is-cons-cluster C and p ∈ C
  shows R ∩ C ≠ {} using assms
  proof (induct)
  case (1 p R)
  obtain Sl where Sl ∈ slices p and Sl ⊆ C using cons-cluster-has-cons-cluster-slice
    using 1.prems by blast
    moreover have Sl ⊆ W using assms(2) calculation(2) is-cons-cluster-def by auto
    ultimately show ?case using 1.hyps assms(2) by fastforce
  qed

Now we show that if p ∈ C, C is a consensus cluster, and quorum Q is such
that Q ∩ C ≠ {}, then Q ∩ W blocks p.
We start by defining the set of participants reachable from a participant
through correct participants. Their union trivially forms a quorum. More-
over, if p is not blocked by a set R, then we show that the set of participants
reachable from p and not blocked by R forms a quorum disjoint from R. It
follows that if p is a member of a consensus cluster C and Q is a quorum of
a member of C, then Q ∩ W must block p, as otherwise quorum intersection
would be violated.

inductive not-blocked for p R where
  [ Sl ∈ slices p; ∀ q ∈ Sl\n  . q /∈ R ∧ ¬blocking-min R q; q ∈ Sl ] ⇒ not-blocked p R q
  p R q
  [| (not-blocked p R p'; p' ∈ W; Sl ∈ slices p'; ∀ q ∈ Sl\n  . q /∈ R ∧ ¬blocking-min R q; q ∈ Sl] ⇒ not-blocked p R q
inductive-cases not-blocked-cases:not-blocked p R q

lemma l4:
  fixes Q p R
  defines Q ≡ { q . not-blocked p R q}
  shows quorum Q
  proof
  have ∃ S ∈ slices n . S ⊆ Q if n∈Q∩W for n
  proof
  have not-blocked p R n using assms that by blast
  hence n /∈ R and ¬blocking-min R n by (metis Int-iff not-blocked.simps that)+
  thus ?thesis using blocking-min.intros not-blocked.intros(2) that unfolding
Q-def by (simp; smt Ball-Collect)
qed
thus thesis by (simp add: quorum-def)
qed

lemma l5:
fixes \( Q \) \( p \) \( R \)
defines \( Q \equiv \{ q . \text{not-blocked} p R q \} \)
assumes \( \neg \text{blocking-min} R p \) \( \text{and} \ p \in C \) \( \text{and} \ \langle \text{is-cons-cluster} C \rangle \)
shows \( \text{quorum-of} p \) \( Q \)
proof
  have \( p \in W \)
  using assms(3,4) cons-cluster-wb by blast
obtain \( Sl \) where \( Sl \in \text{slices} p \) \( \text{and} \ \forall q \in Sl \cap W . \ q \notin R \land \neg \text{blocking-min} R q \)
  by (meson \( p \in W \) assms(2) blocking-min.intros)
hence \( Sl \subseteq Q \)
  unfolding Q-def using not-blocked.intro(1)
  by blast
with l4 \( Sl \in \text{slices} p \) show \( \text{quorum-of} p \) \( Q \)
  using Q-def quorum-of-def by blast
qed

lemma cons-cluster-ne-slices:
assumes \( \text{is-cons-cluster} C \) \( \text{and} \ p \in C \) \( \text{and} \ Sl \in \text{slices} p \)
shows \( Sl \neq \{\} \)
using assms cons-cluster-has-ne-slices cons-cluster-wb stellar_quorum-of-def stellar-axioms
by fastforce

lemma l6:
fixes \( Q \) \( p \) \( R \)
defines \( Q \equiv \{ q . \text{not-blocked} p R q \} \)
shows \( Q \cap R \cap W = \{\} \)
proof
  have \( q \notin R \) if \( \text{not-blocked} p R q \) \( \text{and} \ q \in W \) for \( q \) using that
  by (metis Int-iff not-blocked.simps)
  thus thesis unfolding Q-def by auto
qed

theorem quorum-blocks-cons-cluster:
assumes \( \text{quorum-of-set} C \) \( Q \) \( \text{and} \ p \in C \) \( \text{and} \ \text{is-cons-cluster} C \)
shows blocking-min \( (Q \cap W) p \)
proof (rule contr)
  assume \( \neg \text{blocking-min} (Q \cap W) p \)
  have \( p \in W \)
  using assms(2,3) is-cons-cluster-def by auto
  define \( Q' \) \( \equiv \{ q . \text{not-blocked} p (Q\cap W) q \} \)
  have quorum-of \( p \) \( Q' \)
  using Q'-def \( \neg \text{blocking-min} (Q \cap W) p \)
  using assms(2)
  by blast
  moreover have \( Q' \cap W = \{\} \)
  using Q'-def l6 by fastforce
  ultimately show False
  using assms unfolding is-cons-cluster-def
  by (metis Int-commute inf-sup-aci(2) quorum-of-set-def)
3 Reachability through a set

Here we define the part of a quorum \( Q \) of \( p \) that is reachable through correct participants from \( p \). We show that if \( p \) and \( p' \) are members of the same consensus cluster and \( Q \) is a quorum of \( p \) and \( Q' \) is a quorum of \( p' \), then the intersection \( Q \cap Q' \cap W \) is reachable from both \( p \) and \( p' \) through the consensus cluster.

\[
\text{inductive reachable-through for } p S \text{ where }
  \begin{align*}
  & \text{reachable-through } p S p \\
  & \text{[[reachable-through } p S p'; p' \in W; S \subseteq S; p'' \in S]] \implies \text{reachable-through } p S p''
  \end{align*}
\]

definition truncation where truncation \( p S \equiv \{p'. \text{reachable-through } p S p'\}\)

lemma l13:
  assumes quorum-of \( p Q \) and \( p \in W \) and reachable-through \( p Q p' \)
  shows quorum-of \( p' Q \)
  using assms using quorum-assm reachable-through.cases
  by (metis is-pbqs subset-iff)

lemma l14:
  assumes quorum-of \( p Q \) and \( p \in W \)
  shows quorum \( (\text{truncation } p Q) \)
  proof 
    have \( \exists S \in \text{slices } p' . \forall q \in S . \text{reachable-through } p Q q \) if \( \text{reachable-through } p Q p' \) and \( p' \in W \) for \( p' \)
      by (meson assms l13 quorum-is-quorum-of-some-slice stellar.reachable-through.intros(2) stellar-axioms that)
      thus \(?thesis\)
      by (metis IntE mem-Collect-eq stellar.quorum-def stellar-axioms subsetI truncation-def)
  qed

lemma l15:
  assumes is-cons-cluster \( I \) and quorum-of \( p Q \) and quorum-of \( p' Q' \) and \( p \in I \) and \( p' \in I \) and \( Q \cap Q' \cap W \neq \{\} \)
  shows \( W \cap (\text{truncation } p Q) \cap (\text{truncation } p' Q') \neq \{\} \)
  proof 
    have quorum \( \text{truncation } p Q \) and quorum \( \text{truncation } p' Q' \) using l14 assms is-cons-cluster-def by auto
    moreover have quorum-of-set \( I \) \( (\text{truncation } p Q) \) and quorum-of-set \( I \) \( (\text{truncation } p' Q') \)
      by (metis IntI assms(4,5) calculation mem-Collect-eq quorum-def quorum-of-def quorum-of-set-def reachable-through.intros(1) truncation-def)+
    moreover note is-cons-cluster \( I \)
    ultimately show \(?thesis\) unfolding is-cons-cluster-def by auto
4 Elementary quorums

In this section we define the notion of elementary quorum, which is a quorum that has no strict subset that is a quorum. It follows directly from the definition that every finite quorum contains an elementary quorum. Moreover, we show that if $Q$ is an elementary quorum and $n_1$ and $n_2$ are members of $Q$, then $n_2$ is reachable from $n_1$ in the directed graph over participants defined as the set of edges $(n, m)$ such that $m$ is a member of a slice of $n$. This lemma is used in the companion paper to show that probabilistic leader-election is feasible.

```
locale elementary = stellar
begin

definition elementary where
  elementary s ≡ quorum s ∧ (∀ s′. s′ ⊂ s −→ ¬quorum s′)

lemma finite-subset-wf:
  shows uf {(X, Y). X ⊆ Y ∧ finite Y}
  by (metis finite-psubset-def uf-finite-psubset)

lemma quorum-contains-elementary:
  assumes finite s and ¬elementary s and quorum s
  shows ∃ s′. s′ ⊂ s ∧ elementary s′ using assms
proof (induct s rule:wf-induct[where ?r= {(X, Y). X ⊆ Y ∧ finite Y}])
  case 1
  then show ?case using finite-subset-wf by simp
next
  case (2 x)
  then show ?case
  by (metis (full-types) elementary-def finite-psubset-def finite-subset in-finite-psubset less-le psubset-trans)
qed

inductive path where
  path []
| ∃ x . path [x]
| ∃ l n . [path l; S ∈ Q (hd l); n ∈ S] −→ path (n#l)

theorem elementary-connected:
  assumes elementary s and n_1 ∈ s and n_2 ∈ s and n_1 ∈ W and n_2 ∈ W
  shows ∃ l . hd (rev l) = n_1 ∧ hd l = n_2 ∧ path l (is ?P)
proof
  { assume ¬?P

qed
end
```
define $x$ where $x \equiv \{ n \in s . \exists \ l . \ l \neq [] \land \text{hd (rev } l) = n_1 \land \text{hd } l = n \land \text{path } l \}$

have $n_2 \notin x$ using ($\neg \ ?P$) x-def by auto

have $n_1 \in x$ unfolding x-def using assms(2) pathintros(2) by force

have quorum $x$

proof -
  { fix $n$
    assume $n \in x$
    have $\exists \ S . \ S \in \text{slices } n \land S \subseteq x$
    proof -
      obtain $S$ where $S \in \text{slices } n \land S \subseteq s$ using (elementary s) (n \in x)
    unfolding x-def by (force simp add: elementary-def quorum-def)
    have $S \subseteq x$
    proof -
      { assume $\neg S \subseteq x$
        obtain $m$ where $m \in S$ and $m \notin x$ using ($\neg S \subseteq x$) by auto
        obtain $l'$ where $\text{hd (rev } l') = n_1$ and $\text{hd } l' = n$ and $\text{path } l'$ and $l' \neq []$ using ($n \in x$) x-def by blast
        have $\text{path } (m \# l')$ using (path l') ($m \in S$) ($S \in \text{slices } n$) ($\text{hd } l' = n$) using pathintros(3) by fastforce
        moreover have $\text{hd (rev } (m \# l')) = n_1$ using ($\text{hd (rev } l') = n_1$) ($l'$ $\neq []$) by auto
        ultimately have $m \in x$ using ($m \in S$) ($S \subseteq s$) x-def by auto
        hence False using ($m \notin x$) by blast }
      thus $\neg \text{thesis}$ by blast
    qed
    thus $\neg \text{thesis}$ using ($S \in \text{slices } n$) by blast
  qed }
  thus $\neg \text{thesis}$ by (meson Int-iff quorum-def)
  qed

moreover have $x \subseteq s$
  using ($n_2 \notin x$) assms(3) x-def by blast

ultimately have False using (elementary s)
  using elementary-def by auto
}

thus $\neg \ ?P$ by blast
qed

end

end
References
