Some Properties of Federated Byzantine Agreement Systems

Giuliano Losa

September 11, 2019

Contents

1 Definition of a Federated Byzantine Agreement System

theory *FBA*
imports *Main*
begin

definition *project* **where**
$project\ slices\ S\ n \equiv \{Sl \cap S \mid Sl\ .\ Sl \in slices\ n\}$
— Projecting on S is the same as deleting the complement of S, where deleting is understood as in the Stellar Whitepaper.

locale *FBAS* =
fixes $slices :: 'node \Rightarrow 'node\ set\ set$ — the quorum slices
and $W :: 'node\ set$ — the well-behaved nodes
assumes *slices-ne*: $\bigwedge p\ .\ p \in W \Longrightarrow slices\ p \neq \{\}$ — the set of slices of a well-behaved node is not empty
begin

definition *quorum* **where**
$quorum\ Q \equiv \forall\ p \in Q \cap W\ .\ (\exists\ Sl \in slices\ p\ .\ Sl \subseteq Q)$
— A quorum is a set whose well-behaved members have at least one slice included in the set.

end

2 Intact and the Cascade Theorem

locale *intact* = — Here we fix an intact set I and prove the cascade theorem.
orig:*FBAS slices W*
+ *proj*:*FBAS project slices I W* — We consider the projection of the system on I.
for *slices W I* +
assumes *intact-wb*:$I \subseteq W$ — An intact set is a set I satisfying those three assumptions.
and *q-avail*:*orig.quorum I* — I is a quorum in the original system.
and *q-inter*:$\bigwedge$ Q Q' . ⟦*proj.quorum Q*; *proj.quorum Q'*; $Q \cap I \neq \{\}$; $Q' \cap I \neq \{\}$⟧ $\Longrightarrow Q \cap Q' \cap I \neq \{\}$
— Any two sets that intersect I and that are quorums in the projected system intersect in I. Note that requiring that $Q \cap Q' \neq \{\}$ instead of $Q \cap Q' \cap I \neq \{\}$ would be equivalent.
begin

theorem *blocking-safe*: — A set that blocks an intact node contains an intact node. If this were not the case, quorum availability would trivially be violated.
fixes *S n*
assumes $n \in I$ **and** $\forall$ *Sl*$\in$ *slices n* .*Sl*$\cap$*S* $\neq \{\}$
shows $S \cap I \neq \{\}$
using *assms q-avail intact-wb* **unfolding** *orig.quorum-def*
by *auto* (*metis inf.absorb-iff2 inf-assoc inf-bot-right inf-sup-aci*(*1*))

theorem *cascade*:
— If U is a quorum of an intact node and S is a super-set of U, then either S includes all intact nodes or there is an intact node outside of S which is blocked by the intact members of S. This shows that, in SCP, once the intact members of a quorum accept a statement, a cascading effect occurs and all intact nodes eventually accept it regardless of what befouled and faulty nodes do.
fixes *U S*
assumes *orig.quorum U* **and** $U \cap I \neq \{\}$ **and** $U \subseteq S$
obtains $I \subseteq S$ | $\exists$ $n \in I - S$. $\forall$ *Sl* $\in$ *slices n* . *Sl* $\cap S \cap I \neq \{\}$
proof −
have *False* **if** *1*:$\forall$ $n \in I - S$. $\exists$ *Sl* $\in$ *slices n* . *Sl* $\cap S \cap I = \{\}$ **and** *2*:$\neg(I \subseteq S)$
proof −

First we show that $I - S$ is a quorum in the projected system. This is immediate from the definition of quorum and assumption 1.

have *proj.quorum* $(I-S)$ **using** *1*
unfolding *proj.quorum-def project-def*
by (*auto*; *smt DiffI Diff-Compl Diff-Int-distrib Diff-eq Diff-eq-empty-iff Int-commute*)

Then we show that U is also a quorum in the projected system:

moreover have *proj.quorum U* **using** ‹*orig.quorum U*›
unfolding *proj.quorum-def orig.quorum-def project-def*
by (*simp*; *meson Int-commute inf.coboundedI2*)

Since quorums of I must intersect, we get a contradiction:

ultimately show *False* **using** ‹$U \subseteq S$› ‹$U \cap I \neq \{\}$› ‹$\neg(I{\subseteq}S)$› *q-inter* **by** *auto*
qed
thus *?thesis* **using** *that* **by** *blast*
qed

end

3 The Union Theorem

Here we prove that the union of two intact sets that intersect is intact. This implies that maximal intact sets are disjoint.

locale *intersecting-intact* =
i1:*intact slices* W I_1 + *i2*:*intact slices* W I_2 — We fix two intersecting intact sets I_1 and I_2.
+ *proj*:*FBAS project slices* $(I_1{\cup}I_2)$ W — We consider the projection of the system on $I_1 \cup I_2$.
for *slices* W I_1 I_2 +
assumes *inter*:$I_1 \cap I_2 \neq \{\}$
begin

theorem *union-quorum*: *i1.orig.quorum* $(I_1{\cup}I_2)$ — $I_1 \cup I_2$ is a quorum in the original system.
using *i1.intact-axioms i2.intact-axioms*
unfolding *intact-def FBAS-def intact-axioms-def i1.orig.quorum-def*
by (*metis Int-iff Un-iff le-supI1 le-supI2*)

theorem *union-quorum-intersection*:
assumes *proj.quorum* Q_1 **and** *proj.quorum* Q_2 **and** $Q_1 \cap (I_1{\cup}I_2) \neq \{\}$ **and** $Q_2 \cap (I_1{\cup}I_2) \neq \{\}$
shows $Q_1 \cap Q_2 \cap (I_1{\cup}I_2) \neq \{\}$
— Any two sets that intersect $I_1 \cup I_2$ and that are quorums in the system projected on $I_1 \cup I_2$ intersect in $I_1 \cup I_2$.
proof −

First we show that Q_1 and Q_2 are quorums in the projections on I_1 and I_2.

have *i1.proj.quorum* Q_1 **using** ‹*proj.quorum* Q_1›
unfolding *i1.proj.quorum-def proj.quorum-def project-def*
by *auto* (*metis Int-Un-distrib Int-iff Un-subset-iff*)
moreover have *i2.proj.quorum* Q_2 **using** ‹*proj.quorum* Q_2›
unfolding *i2.proj.quorum-def proj.quorum-def project-def*
by *auto* (*metis Int-Un-distrib Int-iff Un-subset-iff*)
moreover have *i2.proj.quorum* Q_1 **using** ‹*proj.quorum* Q_1›
unfolding *proj.quorum-def i2.proj.quorum-def project-def*
by *auto* (*metis Int-Un-distrib Int-iff Un-subset-iff*)
moreover have *i1.proj.quorum* Q_2 **using** ‹*proj.quorum* Q_2›
unfolding *proj.quorum-def i1.proj.quorum-def project-def*
by *auto* (*metis Int-Un-distrib Int-iff Un-subset-iff*)

Next we show that Q_1 and Q_2 intersect if they are quorums of, respectively, I_1 and I_2. This is the only interesting part of the proof.

moreover have $Q_1 \cap Q_2 \cap (I_1 \cup I_2) \neq \{\}$
if *i1.proj.quorum* Q_1 **and** *i2.proj.quorum* Q_2 **and** *i2.proj.quorum* Q_1
and $Q_1 \cap I_1 \neq \{\}$ **and** $Q_2 \cap I_2 \neq \{\}$
for Q_1 Q_2
proof −
obtain n **where** $n \in I_1 \cap I_2$ **using** *inter* **by** *blast*
have *i1.proj.quorum* I_2
proof −
have *i1.orig.quorum* I_2 **by** (*simp add*: *i2.q-avail*)
thus *?thesis* **unfolding** *i1.orig.quorum-def i1.proj.quorum-def project-def*
by *auto* (*meson Int-commute Int-iff inf-le2 subset-trans*)
qed
moreover note ‹*i1.proj.quorum* Q_1›
ultimately have $Q_1 \cap I_2 \cap I_1 \neq \{\}$ **using** *i1.q-inter inter* ‹$Q_1 \cap I_1 \neq \{\}$› **by** *blast*
moreover note ‹*i2.proj.quorum* Q_2›
moreover note ‹*i2.proj.quorum* Q_1›
ultimately have $Q_1 \cap Q_2 \cap I_2 \neq \{\}$ **using** *i2.q-inter* ‹$Q_2 \cap I_2 \neq \{\}$› **by** *blast*
thus *?thesis* **by** (*simp add*: *inf-sup-distrib1*)
qed

Next we show that Q_1 and Q_2 intersect if they are quorums of the same intact set. This is obvious.

moreover
have $Q_1 \cap Q_2 \cap (I_1 \cup I_2) \neq \{\}$
if *i1.proj.quorum* Q_1 **and** *i1.proj.quorum* Q_2 **and** $Q_1 \cap I_1 \neq \{\}$ **and** $Q_2 \cap I_1 \neq \{\}$
for Q_1 Q_2
by (*simp add*: *Int-Un-distrib i1.q-inter that*)
moreover
have $Q_1 \cap Q_2 \cap (I_1 \cup I_2) \neq \{\}$
if *i2.proj.quorum* Q_1 **and** *i2.proj.quorum* Q_2 **and** $Q_1 \cap I_2 \neq \{\}$ **and** $Q_2 \cap I_2 \neq \{\}$
for Q_1 Q_2
by (*simp add*: *Int-Un-distrib i2.q-inter that*)

Finally we have covered all the cases and get the final result:

ultimately
show *?thesis*
by (*smt Int-Un-distrib Int-commute assms*(*3*,*4*) *sup-bot.right-neutral*)
qed

end

end