Formalization of a Generalized Protocol for Clock Synchronization in Isabelle/HOL

Alwen Tiu
LORIA - http://qsl.loria.fr

March 17, 2025

Abstract

We formalize the generalized Byzantine fault-tolerant clock synchronization protocol of Schneider. This protocol abstracts from particular algorithms or implementations for clock synchronization. This abstraction includes several assumptions on the behaviors of physical clocks and on general properties of concrete algorithms/implementations. Based on these assumptions the correctness of the protocol is proved by Schneider. His proof was later verified by Shankar using the theorem prover EHDM (precursor to PVS). Our formalization in Isabelle/HOL is based on Shankar's formalization.

Contents

1 Introduction

In certain distributed systems, e.g., real-time process-control systems, the existence of a reliable global time source is critical in ensuring the correct functioning of the systems. This reliable global time source can be implemented using several physical clocks distributed on different nodes in the distributed system. Since physical clocks are by nature constantly drifting away from the "real time" and different clocks can have different drift rates, in such a scheme, it is important that these clocks are regularly adjusted so that they are closely synchronized within a certain application-specific safe bound. The design and verification of clock synchronization protocols are often complicated by the additional requirement that the protocols should work correctly under certain types of errors, e.g., failure of some clocks, error in communication network or corrupted messages, etc.

There has been a number of fault-tolerant clock synchronization algorithms studied in the literature, e.g., the *Interactive Convergence Algorithm* (ICA) by Lamport and Melliar-Smith [1], the Lundelius-Lynch algorithm [2], etc., each with its own degree of fault tolerance. One important property that must be satisfied by a clock synchronization algorithm is the agreement property, i.e., at any time t, the difference of the clock readings of any two non-faulty processes must be bounded by a constant (which is fixed according to the domain of applications). At the core of these algorithms is the convergence function that calculates the adjustment to a clock of a process, based on the clock readings of all other processes. Schneider [3] gives an abstract characterization of a wide range of clock synchronization algorithms (based on the convergence functions used) and proves the agreement property in this abstract framework. Schneider's proof was later verified by Shankar [4] in the theorem prover EHDM (precursor to PVS), where eleven axioms about clocks are explicitly stated.

We formalize Schneider's proof in Isabelle/HOL, making use of Shankar's formulation of the clock axioms. The particular formulation of axioms on clock conditions and the statements of the main theorems here are essentially those of Shankar's [4], with some minor changes in syntax. For the full description of the protocol, the general structure of the proof and the meaning of the constants and function symbols used in this formalization, we refer readers to [4].

Acknowledgment I would like to thank Stephan Merz and Pascal Fontaine for useful tips on using Isabelle and particularly the Isar proof language.

2 Isar proof scripts

theory *GenClock* **imports** *Complex-Main* **begin**

2.1 Types and constants definitions

Process is represented by natural numbers. The type 'event' corresponds to synchronization rounds.

type-synonym *process* = *nat*
type-synonym *event* = *nat*
type-synonym *time* = *real*
type-synonym *Clocktime* = *real*

axiomatization
δ :: *real* **and**
μ :: *real* **and**
ϱ :: *real* **and**
rmin :: *real* **and**
rmax :: *real* **and**
β :: *real* **and**
Λ :: *real* **and**

np :: *process* **and**
maxfaults :: *process* **and**

PC :: [*process*, *time*] $\Rightarrow$ *Clocktime* **and**

VC :: [*process*, *time*] $\Rightarrow$ *Clocktime* **and**

te :: [*process*, *event*] $\Rightarrow$ *time* **and**

ϑ :: [$process$, $event$] $\Rightarrow$ ($process \Rightarrow Clocktime$) **and**

IC :: [$process$, $event$, $time$] $\Rightarrow$ $Clocktime$ **and**

$correct$ :: [$process$, $time$] $\Rightarrow$ $bool$ **and**

cfn :: [$process$, ($process \Rightarrow Clocktime$)] $\Rightarrow$ $Clocktime$ **and**

π :: [$Clocktime$, $Clocktime$] $\Rightarrow$ $Clocktime$ **and**

α :: $Clocktime \Rightarrow Clocktime$

definition
$count$ :: [$process \Rightarrow bool$, $process$] $\Rightarrow$ nat **where**
$count\ f\ n = card\ \{p.\ p < n \wedge f\ p\}$

definition

Adj :: [$process$, $event$] $\Rightarrow$ $Clocktime$ **where**
$Adj = (\lambda\ p\ i.\ if\ 0 < i\ then\ cfn\ p\ (\vartheta\ p\ i) - PC\ p\ (te\ p\ i)$
$else\ 0)$

definition

$okRead1$ :: [$process \Rightarrow Clocktime$, $Clocktime$, $process \Rightarrow bool$] $\Rightarrow$ $bool$ **where**
$okRead1\ f\ x\ ppred \longleftrightarrow (\forall\ l\ m.\ ppred\ l \wedge ppred\ m \longrightarrow |f\ l - f\ m| \leq x)$

definition
$okRead2$ :: [$process \Rightarrow Clocktime$, $process \Rightarrow Clocktime$, $Clocktime$,
$process \Rightarrow bool$] $\Rightarrow$ $bool$ **where**
$okRead2\ f\ g\ x\ ppred \longleftrightarrow (\forall\ p.\ ppred\ p \longrightarrow |f\ p - g\ p| \leq x)$

definition
$rho\text{-}bound1$ :: [[$process$, $time$] $\Rightarrow$ $Clocktime$] $\Rightarrow$ $bool$ **where**
$rho\text{-}bound1\ C \longleftrightarrow (\forall\ p\ s\ t.\ correct\ p\ t \wedge s \leq t \longrightarrow C\ p\ t - C\ p\ s \leq (t - s)*(1 + \varrho))$
definition
$rho\text{-}bound2$ :: [[$process$, $time$] $\Rightarrow$ $Clocktime$] $\Rightarrow$ $bool$ **where**
$rho\text{-}bound2\ C \longleftrightarrow (\forall\ p\ s\ t.\ correct\ p\ t \wedge s \leq t \longrightarrow (t - s)*(1 - \varrho) \leq C\ p\ t - C\ p\ s)$

2.2 Clock conditions

Some general assumptions

axiomatization where
$constants\text{-}ax$: $0 < \beta \wedge 0 < \mu \wedge 0 < rmin$
$\wedge\ rmin \leq rmax \wedge 0 < \varrho \wedge 0 < np \wedge maxfaults \leq np$

axiomatization where
$PC\text{-}monotone$: $\forall\ p\ s\ t.\ correct\ p\ t \wedge s \leq t \longrightarrow PC\ p\ s \leq PC\ p\ t$

axiomatization where
$VClock$: $\forall\ p\ t\ i.\ correct\ p\ t \wedge te\ p\ i \leq t \wedge t < te\ p\ (i + 1) \longrightarrow VC\ p\ t = IC\ p\ i\ t$

axiomatization where
IClock: $\forall\ p\ t\ i.\ correct\ p\ t \longrightarrow IC\ p\ i\ t = PC\ p\ t + Adj\ p\ i$

Condition 1: initial skew

axiomatization where
init: $\forall\ p.\ correct\ p\ 0 \longrightarrow 0 \leq PC\ p\ 0 \wedge PC\ p\ 0 \leq \mu$

Condition 2: bounded drift

axiomatization where
rate-1: $\forall\ p\ s\ t.\ correct\ p\ t \wedge s \leq t \longrightarrow PC\ p\ t - PC\ p\ s \leq (t - s)*(1 + \varrho)$ **and**
rate-2: $\forall\ p\ s\ t.\ correct\ p\ t \wedge s \leq t \longrightarrow (t - s)*(1 - \varrho) \leq PC\ p\ t - PC\ p\ s$

Condition 3: bounded interval

axiomatization where
rts0: $\forall\ p\ t\ i.\ correct\ p\ t \wedge t \leq te\ p\ (i+1) \longrightarrow t - te\ p\ i \leq rmax$ **and**
rts1: $\forall\ p\ t\ i.\ correct\ p\ t \wedge te\ p\ (i+1) \leq t \longrightarrow rmin \leq t - te\ p\ i$

Condition 4 : bounded delay

axiomatization where
rts2a: $\forall\ p\ q\ t\ i.\ correct\ p\ t \wedge correct\ q\ t \wedge te\ q\ i + \beta \leq t \longrightarrow te\ p\ i \leq t$ **and**
rts2b: $\forall\ p\ q\ i.\ correct\ p\ (te\ p\ i) \wedge correct\ q\ (te\ q\ i) \longrightarrow abs(te\ p\ i - te\ q\ i) \leq \beta$

Condition 5: initial synchronization

axiomatization where
synch0: $\forall\ p.\ te\ p\ 0 = 0$

Condition 6: nonoverlap

axiomatization where
nonoverlap: $\beta \leq rmin$

Condition 7: reading errors

axiomatization where
readerror: $\forall\ p\ q\ i.\ correct\ p\ (te\ p\ (i+1)) \wedge correct\ q\ (te\ p\ (i+1)) \longrightarrow$
$abs(\vartheta\ p\ (i+1)\ q - IC\ q\ i\ (te\ p\ (i+1))) \leq \Lambda$

Condition 8: bounded faults

axiomatization where
correct-closed: $\forall\ p\ s\ t.\ s \leq t \wedge correct\ p\ t \longrightarrow correct\ p\ s$ **and**
correct-count: $\forall\ t.\ np - maxfaults \leq count\ (\lambda\ p.\ correct\ p\ t)\ np$

Condition 9: Translation invariance

axiomatization where
trans-inv: $\forall\ p\ f\ x.\ 0 \leq x \longrightarrow cfn\ p\ (\lambda\ y.\ f\ y + x) = cfn\ p\ f + x$

Condition 10: precision enhancement

axiomatization where
prec-enh:
$\forall\ ppred\ p\ q\ f\ g\ x\ y.$
$np - maxfaults \leq count\ ppred\ np \wedge$
$okRead1\ f\ y\ ppred \wedge okRead1\ g\ y\ ppred \wedge$

$okRead2\ f\ g\ x\ ppred \wedge ppred\ p \wedge ppred\ q$
$\longrightarrow abs(cfn\ p\ f\ -\ cfn\ q\ g) \leq \pi\ x\ y$

Condition 11: accuracy preservation

axiomatization where
acc-prsv:
$\forall\ ppred\ p\ q\ f\ x.\ okRead1\ f\ x\ ppred \wedge np - maxfaults \leq count\ ppred\ np$
$\wedge ppred\ p \wedge ppred\ q \longrightarrow abs(cfn\ p\ f\ -\ f\ q) \leq \alpha\ x$

2.2.1 Some derived properties of clocks

lemma *rts0d*:
assumes *cp*: $correct\ p\ (te\ p\ (i+1))$
shows $te\ p\ (i+1) - te\ p\ i \leq rmax$
⟨*proof*⟩

lemma *rts1d*:
assumes *cp*: $correct\ p\ (te\ p\ (i+1))$
shows $rmin \leq te\ p\ (i+1) - te\ p\ i$
⟨*proof*⟩

lemma *rte*:
assumes *cp*: $correct\ p\ (te\ p\ (i+1))$
shows $te\ p\ i \leq te\ p\ (i+1)$
⟨*proof*⟩

lemma *beta-bound1*:
assumes *corr-p*: $correct\ p\ (te\ p\ (i+1))$
and *corr-q*: $correct\ q\ (te\ p\ (i+1))$
shows $0 \leq te\ p\ (i+1) - te\ q\ i$
⟨*proof*⟩

lemma *beta-bound2*:
assumes *corr-p*: $correct\ p\ (te\ p\ (i+1))$
and *corr-q*: $correct\ q\ (te\ q\ i)$
shows $te\ p\ (i+1) - te\ q\ i \leq rmax + \beta$
⟨*proof*⟩

2.2.2 Bounded-drift for logical clocks (IC)

lemma *bd*:
assumes *ie*: $s \leq t$
and *rb1*: $rho\text{-}bound1\ C$
and *rb2*: $rho\text{-}bound2\ D$
and *PC-ie*: $D\ q\ t - D\ q\ s \leq C\ p\ t - C\ p\ s$
and *corr-p*: $correct\ p\ t$
and *corr-q*: $correct\ q\ t$
shows $|\ C\ p\ t - D\ q\ t\ |\ \leq |\ C\ p\ s - D\ q\ s\ | + 2{*}\varrho{*}(t - s)$
⟨*proof*⟩

lemma *bounded-drift*:
assumes *ie*: $s \leq t$
and *rb1*: $rho\text{-}bound1\ C$

and *rb2*: *rho-bound2 C*
and *rb3*: *rho-bound1 D*
and *rb4*: *rho-bound2 D*
and *corr-p*: *correct p t*
and *corr-q*: *correct q t*
shows $|C\ p\ t - D\ q\ t| \leq |C\ p\ s - D\ q\ s| + 2*\varrho*(t - s)$
⟨*proof*⟩

Drift rate of logical clocks

lemma *IC-rate1*:
rho-bound1 $(\lambda\ p\ t.\ IC\ p\ i\ t)$
⟨*proof*⟩

lemma *IC-rate2*:
rho-bound2 $(\lambda\ p\ t.\ IC\ p\ i\ t)$
⟨*proof*⟩

Auxilary function *ICf*: we introduce this to avoid some unification problem in some tactic of isabelle.

definition
ICf :: *nat* ⇒ (*process* ⇒ *time* ⇒ *Clocktime*) **where**
$ICf\ i = (\lambda\ p\ t.\ IC\ p\ i\ t)$

lemma *IC-bd*:
assumes *ie*: $s \leq t$
and *corr-p*: *correct p t*
and *corr-q*: *correct q t*
shows $|IC\ p\ i\ t - IC\ q\ j\ t| \leq |IC\ p\ i\ s - IC\ q\ j\ s| + 2*\varrho*(t - s)$
⟨*proof*⟩

lemma *event-bound*:
assumes *ie1*: $0 \leq (t::real)$
and *corr-p*: *correct p t*
and *corr-q*: *correct q t*
shows $\exists\ i.\ t < max\ (te\ p\ i)\ (te\ q\ i)$
⟨*proof*⟩

2.3 Agreement property

definition $\gamma 1\ x = \pi\ (2*\varrho*\beta + 2*\Lambda)\ (2*\Lambda + x + 2*\varrho*(rmax + \beta))$
definition $\gamma 2\ x = x + 2*\varrho*rmax$
definition $\gamma 3\ x = \alpha\ (2*\Lambda + x + 2*\varrho*(rmax + \beta)) + \Lambda + 2*\varrho*\beta$

definition
okmaxsync :: [*nat*, *Clocktime*] ⇒ *bool* **where**
okmaxsync i x ⟷ (∀ *p q*. *correct p* (*max* (*te p i*) (*te q i*))
∧ *correct q* (*max* (*te p i*) (*te q i*)) ⟶
$|IC\ p\ i\ (max\ (te\ p\ i)\ (te\ q\ i)) - IC\ q\ i\ (max\ (te\ p\ i)\ (te\ q\ i))| \leq x$)

definition
okClocks :: [*process*, *process*, *nat*] ⇒ *bool* **where**
okClocks p q i ⟷ (∀ *t*. $0 \leq t \wedge t < max\ (te\ p\ i)\ (te\ q\ i)$
∧ *correct p t* ∧ *correct q t*
⟶ $|VC\ p\ t - VC\ q\ t| \leq \delta$)

lemma *okClocks-sym*:
assumes *ok-pq*: *okClocks p q i*
shows *okClocks q p i*
⟨*proof*⟩

lemma *ICp-Suc*:
assumes *corr-p*: *correct p (te p (i+1))*
shows *IC p (i+1) (te p (i+1)) = cfn p (ϑ p (i+1))*
⟨*proof*⟩

lemma *IC-trans-inv*:
assumes *ie1*: *te q (i+1) ≤ te p (i+1)*
and *corr-p*: *correct p (te p (i+1))*
and *corr-q*: *correct q (te p (i+1))*
shows
IC q (i+1) (te p (i+1)) =
  cfn q (λ n. ϑ q (i+1) n + (PC q (te p (i+1)) − PC q (te q (i+1))))
(**is** *?T1 = ?T2*)
⟨*proof*⟩

lemma *beta-rho*:
assumes *ie*: *te q (i+1) ≤ te p (i+1)*
and *corr-p*: *correct p (te p (i+1))*
and *corr-q*: *correct q (te p (i+1))*
and *corr-l*: *correct l (te p (i+1))*
shows *|(PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1))| ≤ β∗ϱ*
⟨*proof*⟩

This lemma (and the next one pe-cond2) proves an assumption used in the precision enhancement.

lemma *pe-cond1*:
assumes *ie*: *te q (i+1) ≤ te p (i+1)*
and *corr-p*: *correct p (te p (i+1))*
and *corr-q*: *correct q (te p (i + 1))*
and *corr-l*: *correct l (te p (i+1))*
shows *|ϑ q (i+1) l + (PC q (te p (i+1)) − PC q (te q (i+1))) −*
        ϑ p (i+1) l| ≤ 2∗ ϱ ∗ β + 2∗Λ
(**is** *?M ≤ ?N*)
⟨*proof*⟩

lemma *pe-cond2*:
assumes *ie*: *te m i ≤ te l i*
and *corr-k*: *correct k (te k (i+1))*
and *corr-l-tk*: *correct l (te k (i+1))*
and *corr-m-tk*: *correct m (te k (i+1))*
and *ind-hyp*: *|IC l i (te l i) − IC m i (te l i)| ≤ δS*
shows *|ϑ k (i+1) l − ϑ k (i+1) m| ≤ 2∗Λ + δS + 2∗ϱ∗(rmax + β)*
⟨*proof*⟩

lemma *theta-bound*:
assumes *corr-l*: *correct l (te p (i+1))*
and *corr-m*: *correct m (te p (i+1))*
and *corr-p*: *correct p (te p (i+1))*

and *IC-bound*:
$|IC\ l\ i\ (max\ (te\ l\ i)\ (te\ m\ i)) - IC\ m\ i\ (max\ (te\ l\ i)\ (te\ m\ i))|$
$\le \delta S$
shows $|\vartheta\ p\ (i{+}1)\ l - \vartheta\ p\ (i{+}1)\ m|$
$\le\ 2{*}\Lambda + \delta S + 2{*}\varrho{*}(rmax + \beta)$
⟨*proof*⟩

lemma *four-one-ind-half*:
assumes *ie1*: $\beta \le rmin$
and *ie2*: $\mu \le \delta S$
and *ie3*: $\gamma 1\ \delta S \le \delta S$
and *ind-hyp*: $okmaxsync\ i\ \delta S$
and *ie4*: $te\ q\ (i{+}1) \le te\ p\ (i{+}1)$
and *corr-p*: $correct\ p\ (te\ p\ (i{+}1))$
and *corr-q*: $correct\ q\ (te\ p\ (i{+}1))$
shows $|IC\ p\ (i{+}1)\ (te\ p\ (i{+}1)) - IC\ q\ (i{+}1)\ (te\ p\ (i{+}1))| \le \delta S$
⟨*proof*⟩

Theorem 4.1 in Shankar's paper.

theorem *four-one*:
assumes *ie1*: $\beta \le rmin$
and *ie2*: $\mu \le \delta S$
and *ie3*: $\gamma 1\ \delta S \le \delta S$
shows $okmaxsync\ i\ \delta S$
⟨*proof*⟩

lemma *VC-cfn*:
assumes *corr-p*: $correct\ p\ (te\ p\ (i{+}1))$
and *ie*: $te\ p\ (i{+}1) < te\ p\ (i{+}2)$
shows $VC\ p\ (te\ p\ (i{+}1)) = cfn\ p\ (\vartheta\ p\ (i{+}1))$
⟨*proof*⟩

Lemma for the inductive case in Theorem 4.2

lemma *four-two-ind*:
assumes *ie1*: $\beta \le rmin$
and *ie2*: $\mu \le \delta S$
and *ie3*: $\gamma 1\ \delta S \le \delta S$
and *ie4*: $\gamma 2\ \delta S \le \delta$
and *ie5*: $\gamma 3\ \delta S \le \delta$
and *ie6*: $te\ q\ (i{+}1) \le te\ p\ (i{+}1)$
and *ind-hyp*: $okClocks\ p\ q\ i$
and *t-bound1*: $0 \le t$
and *t-bound2*: $t < max\ (te\ p\ (i{+}1))\ (te\ q\ (i{+}1))$
and *t-bound3*: $max\ (te\ p\ i)\ (te\ q\ i) \le t$
and *tpq-bound*: $max\ (te\ p\ i)\ (te\ q\ i) < max\ (te\ p\ (i{+}1))\ (te\ q\ (i{+}1))$
and *corr-p*: $correct\ p\ t$
and *corr-q*: $correct\ q\ t$
shows $|VC\ p\ t - VC\ q\ t| \le \delta$
⟨*proof*⟩

Theorem 4.2 in Shankar's paper.

theorem *four-two*:
assumes *ie1*: $\beta \le rmin$

and *ie2*: $\mu \le \delta S$
and *ie3*: $\gamma 1\ \delta S \le \delta S$
and *ie4*: $\gamma 2\ \delta S \le \delta$
and *ie5*: $\gamma 3\ \delta S \le \delta$
shows *okClocks p q i*
⟨*proof*⟩

The main theorem: all correct clocks are synchronized within the bound delta.

theorem *agreement*:
assumes *ie1*: $\beta \le rmin$
and *ie2*: $\mu \le \delta S$
and *ie3*: $\gamma 1\ \delta S \le \delta S$
and *ie4*: $\gamma 2\ \delta S \le \delta$
and *ie5*: $\gamma 3\ \delta S \le \delta$
and *ie6*: $0 \le t$
and *cpq*: $correct\ p\ t \wedge correct\ q\ t$
shows $|VC\ p\ t - VC\ q\ t| \le \delta$
⟨*proof*⟩

end

References

[1] L. Lamport and P. M. Melliar-Smith. Synchronizing clocks in the presence of faults. *J. ACM*, 32(1):52–78, 1985.

[2] J. Lundelius and N. Lynch. A new fault-tolerant algorithm for clock synchronization. In *Proceedings of PODC '84*, pages 75–88, New York, NY, USA, 1984. ACM Press.

[3] F. B. Schneider. Understanding protocols for byzantine clock synchronization. Technical Report 87-859, Department of Computer Science, Cornell University, August 1987.

[4] N. Shankar. Mechanical verification of a generalized protocol for byzantine fault tolerant clock synchronization. In J. Vytopil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 571 of *LNCS*. Springer Verlag, Jan. 1992.