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

Alwen Tiu

August 16, 2018

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 1

2 Isar proof scripts 2

2.1 Types and constants definitions 2

2.2 Clock conditions 3

2.2.1 Some derived properties of clocks 5

2.2.2 Bounded-drift for logical clocks (IC) 5

2.3 Agreement property 6

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] ⇒ Clocktime and
VC :: [process, time] ⇒ Clocktime and
te :: [process, event] ⇒ time and
```
\( \vartheta :: [\text{process, event}] \Rightarrow (\text{process} \Rightarrow \text{Clocktime}) \) and

\( IC :: [\text{process, event, time}] \Rightarrow \text{Clocktime} \) and

\( \text{correct} :: [\text{process, time}] \Rightarrow \text{bool} \) and

\( \text{cfn} :: [\text{process}, (\text{process} \Rightarrow \text{Clocktime})] \Rightarrow \text{Clocktime} \) and

\( \pi :: [\text{Clocktime}, \text{Clocktime}] \Rightarrow \text{Clocktime} \) and

\( \alpha :: \text{Clocktime} \Rightarrow \text{Clocktime} \)

**definition**

\( \text{count} :: [\text{process} \Rightarrow \text{bool}, \text{process}] \Rightarrow \text{nat} \) where

\( \text{count} f n = \text{card} \{ p. \ p < n \land f p \} \)

**definition**

\( \text{Adj} :: [\text{process, event}] \Rightarrow \text{Clocktime} \) where

\( \text{Adj} = (\lambda p i. \ \text{if } 0 < i \ \text{then} \ \text{cfn p} (\vartheta p i) - \text{PC p} (\text{te p} i) \ \text{else} \ 0) \)

**definition**

\( \text{okRead1} :: [\text{process} \Rightarrow \text{Clocktime}, \text{Clocktime}, \text{process} \Rightarrow \text{bool}] \Rightarrow \text{bool} \) where

\( \text{okRead1} f x p p \text{pred} \longleftrightarrow (\forall l m. \ \text{p p r e d} l \land \text{p p r e d} m \longrightarrow |f l - f m| \leq x) \)

**definition**

\( \text{okRead2} :: [\text{process} \Rightarrow \text{Clocktime}, \text{process} \Rightarrow \text{Clocktime}, \text{Clocktime}, \text{process} \Rightarrow \text{bool}] \Rightarrow \text{bool} \) where

\( \text{okRead2} f g x p p \text{pred} \longleftrightarrow (\forall p. \ \text{p p r e d} p \longrightarrow |f p - g p| \leq x) \)

**definition**

\( \text{rho-bound1} :: [\text{process, time}] \Rightarrow \text{Clocktime} \) \Rightarrow \text{bool} \) where

\( \text{rho-bound1} C \longleftrightarrow (\forall p s t. \ \text{c o r r e c t p} t \land s \leq t \longrightarrow \text{C p t} - \text{C p s} \leq (t - s) \ast (1 + \varrho)) \)

**definition**

\( \text{rho-bound2} :: [\text{process, time}] \Rightarrow \text{Clocktime} \) \Rightarrow \text{bool} \) where

\( \text{rho-bound2} C \longleftrightarrow (\forall p s t. \ \text{c o r r e c t p} t \land s \leq t \longrightarrow (t - s) \ast (1 - \varrho) \leq \text{C p t} - \text{C p s}) \)

### 2.2 Clock conditions

Some general assumptions

**axiomatization where**

\( \text{constants-ax: } 0 < \beta \land 0 < \mu \land 0 < r \min \land r \min \leq r \max \land 0 < \varrho \land 0 < n \p \land \max f a u l t s \leq n \p \)

**axiomatization where**

\( \text{PC-monotone: } \forall p s t. \ \text{c o r r e c t p} t \land s \leq t \longrightarrow \text{PC} p s \leq \text{PC} p t \)

**axiomatization where**

\( \text{VClock: } \forall p t i. \ \text{c o r r e c t p} t \land \text{te} p i \leq t \land t < \text{te} p (i + 1) \longrightarrow \text{VC p} t = \text{IC p} t \)

3
axiomatization where

IClock: ∀ p t i. correct p t → IC p i t = PC p t + Adj p i

Condition 1: initial skew

axiomatization where

init: ∀ p. correct p 0 → 0 ≤ PC p 0 ∧ PC p 0 ≤ μ

Condition 2: bounded drift

axiomatization where

rate-1: ∀ p s t. correct p t ∧ s ≤ t → PC p t - PC p s ≤ (t - s)*(t + g) and
rate-2: ∀ p s t. correct p t ∧ s ≤ t → (t - s)*(1 - g) ≤ PC p t - PC p s

Condition 3: bounded interval

axiomatization where

rts0: ∀ p t i. correct p t ∧ t ≤ te p (i+1) → t - te p i ≤ rmax and
rts1: ∀ p t i. correct p t ∧ te p (i+1) ≤ t → rmin ≤ t - te p i

Condition 4: bounded delay

axiomatization where

rts2a: ∀ p q t i. correct p t ∧ correct q t ∧ te q i + β ≤ t → te p i ≤ t and
rts2b: ∀ p q i. correct p (te p i) ∧ correct q (te q i) → abs(te p i - te q i) ≤ β

Condition 5: initial synchronization

axiomatization where

synch0: ∀ p. te p 0 = 0

Condition 6: nonoverlap

axiomatization where

nonoverlap: β ≤ rmin

Condition 7: reading errors

axiomatization where

readerror: ∀ p q i. correct p (te p (i+1)) ∧ correct q (te p (i+1)) →
abs(ϑ p (i+1) q - IC q i (te p (i+1))) ≤ Λ

Condition 8: bounded faults

axiomatization where

correct-closed: ∀ p s t. s ≤ t ∧ correct p t → correct p s and
correct-count: ∀ t. np - maxfaults ≤ count (λ p. correct p t) np

Condition 9: Translation invariance

axiomatization where

trans-inv: ∀ p f x. 0 ≤ x → cfn p (λ y. f y + x) = cfn p f + x

Condition 10: precision enhancement

axiomatization where

prec-enh:
∀ ppred p q f g x y.
np - maxfaults ≤ count ppred np ∧
okRead1 f y ppred ∧ okRead1 g y ppred ∧
\[
okRead2 \ f \ g \ x \ ppred \land ppred \land ppred \ q \\
\rightarrow abs(cfn \ p \ f - cfn \ q \ g) \leq \pi \ x \ y
\]

Condition 11: accuracy preservation

axiomatization where

\[
\forall \ ppred \ p \ q \ f \ x. \ okRead1 \ f \ x \ ppred \land np - maxfaults \leq count \ ppred \ np \\
\land ppred \ p \land ppred \ q \rightarrow abs(cfn \ p \ f - f \ q) \leq \alpha \ x
\]

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-bound1 C
and rb2: rho-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 \rho* (t - s)
-proof

lemma bounded-drift:
assumes ie: s \leq t
and rb1: rho-bound1 C
and \( \text{rb}2: \rho\text{-bound}2 \ C \)
and \( \text{rb}3: \rho\text{-bound}1 \ D \)
and \( \text{rb}4: \rho\text{-bound}2 \ D \)
and \( \text{corr-p: correct} \ p \ t \)
and \( \text{corr-q: correct} \ q \ t \)
shows \( |C \ p \ t - D \ q \ t| \leq |C \ p \ s - D \ q \ s| + 2*\rho*(t - s) \)
\( \langle \text{proof} \rangle \)

Drift rate of logical clocks

\textbf{lemma IC-rate1:}
\( \rho\text{-bound}1 \ (\lambda \ p \ t. \ IC \ p \ i \ t) \)
\( \langle \text{proof} \rangle \)

\textbf{lemma IC-rate2:}
\( \rho\text{-bound}2 \ (\lambda \ p \ t. \ IC \ p \ i \ t) \)
\( \langle \text{proof} \rangle \)

\textbf{Auxilary function ICf:} we introduce this to avoid some unification problem in some tactic of isabelle.

\textbf{definition}
\( ICf :: \text{nat} \Rightarrow (\text{process} \Rightarrow \text{time} \Rightarrow \text{Clocktime}) \)
\where
\( ICf \ i = (\lambda \ p \ t. \ IC \ p \ i \ t) \)

\textbf{lemma IC-bd:}
\textbf{assumes} \( ie: \ s \leq t \)
and \( \text{corr-p: correct} \ p \ t \)
and \( \text{corr-q: correct} \ q \ t \)
shows \( |IC \ p \ i \ t - IC \ q \ j \ t| \leq |IC \ p \ i \ s - IC \ q \ j \ s| + 2*\rho*(t - s) \)
\( \langle \text{proof} \rangle \)

\textbf{lemma event-bound:}
\textbf{assumes} \( ie1: \ 0 \leq (t::\text{real}) \)
and \( \text{corr-p: correct} \ p \ t \)
and \( \text{corr-q: correct} \ q \ t \)
shows \( \exists \ i. \ t < \max (\text{te} \ p \ i) (\text{te} \ q \ i) \)
\( \langle \text{proof} \rangle \)

\subsection*{2.3 Agreement property}

\textbf{definition} \( \gamma 1 \ x = \pi \ (2*\rho*\beta + 2*\Lambda) \ (2*\Lambda + x + 2*\rho*(r_{\text{max}} + \beta)) \)

\textbf{definition} \( \gamma 2 \ x = x + 2*\rho*r_{\text{max}} \)

\textbf{definition} \( \gamma 3 \ x = \alpha \ (2*\Lambda + x + 2*\rho*(r_{\text{max}} + \beta)) + \Lambda + 2*\rho*\beta \)

\textbf{definition}
\textbf{okmaxsync :: [nat, Clocktime] \Rightarrow bool where}
\( \text{okmaxsync} \ i \ x \leftarrow \forall \ p \ q. \ \text{correct} \ p \ (\max (\text{te} \ p \ i) (\text{te} \ q \ i)) \)
\( \land \ \text{correct} \ q \ (\max (\text{te} \ p \ i) (\text{te} \ q \ i)) \rightarrow \)
\( |IC \ p \ i \ (\max (\text{te} \ p \ i) (\text{te} \ q \ i)) - IC \ q \ i \ (\max (\text{te} \ p \ i) (\text{te} \ q \ i))| \leq x \)

\textbf{definition}
\textbf{okClocks :: [process, process, nat] \Rightarrow bool where}
\( \text{okClocks} \ p \ q \ i \leftarrow \forall \ t. \ 0 \leq t \land t < \max (\text{te} \ p \ i) (\text{te} \ q \ i) \)
\( \land \ \text{correct} \ p \ t \land \text{correct} \ q \ t \)
\( \rightarrow |VC \ p \ t - VC \ q \ t| \leq \delta \)
**lemma** okClocks-sym:
**assumes** ok-pq: okClocks p q i
**shows** okClocks q p i

**lemma** I Cp-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))

**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))))

**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** |ϑ q (i+1) l + (PC q (te p (i+1)) − PC q (te q (i+1))) − ϑ p (i+1) l| ≤ β*ϱ*

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

**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:

\[ |IC\ i\ (\max\ (te\ l\ i)\ (te\ m\ i)) - IC\ m\ i\ (\max\ (te\ l\ i)\ (te\ m\ i))| \leq \delta S \]

shows \[|\vartheta\ p\ (i+1)\ l - \vartheta\ p\ (i+1)\ m| \leq 2\Lambda + \delta S + 2\Lambda (r_{\max} + \beta)\]

\[\langle\text{proof}\rangle\]

**Lemma four-one-ind-half:**

- Assumes \(ie1: \beta \leq r_{\min}\)
- and \(ie2: \mu \leq \delta S\)
- and \(ie3: \gamma_1 \delta S \leq \delta S\)
- and ind-hyp: \(ok_{\maxsync} i \delta S\)
- and \(ie4: te\ q\ (i+1) \leq te\ p\ (i+1)\)
- and corr-p: \(correct\ p\ (te\ p\ (i+1))\)
- and corr-q: \(correct\ q\ (te\ p\ (i+1))\)
- shows \[|IC\ p\ (i+1)\ (te\ p\ (i+1)) - IC\ q\ (i+1)\ (te\ p\ (i+1))| \leq \delta S\]

\[\langle\text{proof}\rangle\]

Theorem 4.1 in Shankar’s paper.

**Theorem four-one:**

- Assumes \(ie1: \beta \leq r_{\min}\)
- and \(ie2: \mu \leq \delta S\)
- and \(ie3: \gamma_1 \delta S \leq \delta S\)
- and \(ie4: te\ q\ (i+1) \leq te\ p\ (i+1)\)
- and \(corr-p: correct\ p\ (te\ p\ (i+1))\)
- and \(corr-q: correct\ q\ (te\ p\ (i+1))\)
- shows \(ok_{\maxsync} i \delta S\)

\[\langle\text{proof}\rangle\]

**Lemma VC-cfn:**

- Assumes \(corr-p: correct\ p\ (te\ p\ (i+1))\)
- and \(ie: te\ p\ (i+1) < te\ p\ (i+2)\)
- shows \(VC\ p\ (te\ p\ (i+1)) = cfn\ p\ (\vartheta\ p\ (i+1))\)

\[\langle\text{proof}\rangle\]

Lemma for the inductive case in Theorem 4.2

**Lemma four-two-ind:**

- Assumes \(ie1: \beta \leq r_{\min}\)
- and \(ie2: \mu \leq \delta S\)
- and \(ie3: \gamma_1 \delta S \leq \delta S\)
- and \(ie4: \gamma_2 \delta S \leq \delta\)
- and \(ie5: \gamma_3 \delta S \leq \delta\)
- and \(ie6: te\ q\ (i+1) \leq te\ p\ (i+1)\)
- and ind-hyp: \(ok_{\Clocks}\ p\ q\ i\)
- and \(t\text{-bound1}: 0 \leq t\)
- and \(t\text{-bound2}: t < \max\ (te\ p\ (i+1))\ (te\ q\ (i+1))\)
- and \(t\text{-bound3}: \max\ (te\ p\ i)\ (te\ q\ i) \leq t\)
- and \(tpq\text{-bound}: \max\ (te\ p\ i)\ (te\ q\ i) < \max\ (te\ p\ (i+1))\ (te\ q\ (i+1))\)
- and \(corr-p: correct\ p\ t\)
- and \(corr-q: correct\ q\ t\)
- shows \(|VC\ p\ t - VC\ q\ t| \leq \delta\)

\[\langle\text{proof}\rangle\]

Theorem 4.2 in Shankar’s paper.

**Theorem four-two:**

- Assumes \(ie1: \beta \leq r_{\min}\)

8
and \( ie2: \mu \leq \delta S \)
and \( ie3: \gamma_1 \delta S \leq \delta S \)
and \( ie4: \gamma_2 \delta S \leq \delta \)
and \( ie5: \gamma_3 \delta S \leq \delta \)
shows okClocks \( p \ q \ i \)
(proof)

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

theorem agreement:
  assumes \( ie1: \beta \leq r_{\text{min}} \)
  and \( ie2: \mu \leq \delta S \)
  and \( ie3: \gamma_1 \delta S \leq \delta S \)
  and \( ie4: \gamma_2 \delta S \leq \delta \)
  and \( ie5: \gamma_3 \delta S \leq \delta \)
  and \( ie6: 0 \leq t \)
  and \( cpq: \text{correct } p \ t \ \wedge \ \text{correct } q \ t \)
shows \( |VC \ p \ t \ - \ VC \ q \ t| \leq \delta \)
(proof)

end

References


