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

Alwen Tiu

May 28, 2015

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

```isabelle
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.

```
\[ \varnothing : [\text{process}, \text{event}] \Rightarrow (\text{process} \Rightarrow \text{Clocktime}) \text{ and} \]

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

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

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

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

\[ \alpha : \text{Clocktime} \Rightarrow \text{Clocktime} \]

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

\[ \text{definition} \]
\[ \text{Adj} : [\text{process}, \text{event}] \Rightarrow \text{Clocktime} \text{ where} \]
\[ \text{Adj} = (\lambda p \ i. \text{if } 0 < i \text{ then } \text{cfn } p (\varnothing \ p \ i) - \text{PC } p (\text{te } p \ i) \text{ else } 0) \]

\[ \text{definition} \]
\[ \text{okRead1} : [\text{process} \Rightarrow \text{Clocktime}, \text{Clocktime}, \text{process} \Rightarrow \text{bool}] \Rightarrow \text{bool} \text{ where} \]
\[ \text{okRead1 } f \ x \ ppred \longleftrightarrow (\forall \ l \ m. \ ppred l \land ppred m \longrightarrow |f \ l - f \ m| \leq x) \]

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

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

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

### 2.2 Clock conditions

Some general assumptions

\[ \text{axiomatization where} \]
\[ \text{constants-ax: } 0 < \beta \land 0 < \mu \land 0 < r_{\text{min}} \]
\[ \land r_{\text{min}} \leq r_{\text{max}} \land 0 < \varrho \land 0 < np \land \text{maxfaults} \leq np \]

\[ \text{axiomatization where} \]
\[ \text{PC-monotone: } \forall \ p \ s \ t. \text{correct } p \ t \land s \leq t \longrightarrow PC \ p \ s \leq PC \ p \ t \]

\[ \text{axiomatization where} \]
\[ \text{VClock: } \forall \ p \ t \ i. \text{correct } p \ t \land \text{te } p \ i \leq t \land t < \text{te } p (i + 1) \longrightarrow VC \ p \ t = IC \ p \ i \ t \]
axiomatization where
IClock: ∀ p t i. correct p t = IC p t = PC p t + Adj p i

Condition 1: initial skew

axiomatization where
init: ∀ p. correct p 0 = 0 \leq PC p 0 \land PC p 0 \leq \mu

Condition 2: bounded drift

axiomatization where
rate-1: \forall p s t. correct p t \land s \leq t \rightarrow PC p t - PC p s \leq (t - s) \mu
rate-2: \forall p s t. correct p t \land s \leq t \rightarrow (t - s) \mu \leq PC p t - PC p s

Condition 3: bounded interval

axiomatization where
rts0: \forall p t i. correct p t \land te p (i+1) = t - te p i \leq rmax
rts1: \forall p t i. correct p t \land te p (i+1) \leq t \rightarrow rmin \leq t - te p i

Condition 4: bounded delay

axiomatization where
rts2a: \forall p q t i. correct p t \land correct q t \land te q i + \beta \leq t \rightarrow te p i \leq t
rts2b: \forall p q t i. correct p (te p i) \land correct q (te q i) \rightarrow 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)) \land correct q (te p (i+1)) \rightarrow abs(\theta 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. \land correct p t \rightarrow correct p s
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 \rightarrow 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 \land okRead1 f y ppred \land okRead1 g y ppred \land
\[ \text{okRead2 } f \; g \; x \; \text{ppred} \land \text{ppred } p \land \text{ppred } q \]

\[ \rightarrow \text{abs}(\text{cfn } p \; f - \; \text{cfn } q \; g) \leq \pi \; x \; y \]

Condition 11: accuracy preservation

axiomatization where

\[ \text{acc-prsv:} \]

\[ \forall \; \text{ppred } p \; q \; f \; x. \; \text{okRead1 } f \; x \; \text{ppred} \land \; \text{np} - \; \text{maxfaults} \leq \; \text{count } \text{ppred } \text{np} \]

\[ \land \; \text{ppred } p \land \; \text{ppred } q \rightarrow \text{abs}(\text{cfn } p \; f - \; f \; q) \leq \alpha \; x \]

2.2.1 Some derived properties of clocks

lemma \text{rts0d}:

assumes \text{cp: correct } p \; (\text{te } p \; (i+1))

shows \text{te } p \; (i+1) - \; \text{te } p \; i \leq \text{rmax}

(proof)

lemma \text{rts1d}:

assumes \text{cp: correct } p \; (\text{te } p \; (i+1))

shows \text{rmin} \leq \text{te } p \; (i+1) - \; \text{te } p \; i

(proof)

lemma \text{rte}:

assumes \text{cp: correct } p \; (\text{te } p \; (i+1))

shows \text{te } p \; i \leq \text{te } p \; (i+1)

(proof)

lemma \text{beta-bound1}:

assumes \text{corr-p: correct } p \; (\text{te } p \; (i+1))

and \text{corr-q: correct } q \; (\text{te } p \; (i+1))

shows \text{0} \leq \text{te } p \; (i+1) - \; \text{te } q \; i

(proof)

lemma \text{beta-bound2}:

assumes \text{corr-p: correct } p \; (\text{te } p \; (i+1))

and \text{corr-q: correct } q \; (\text{te } q \; i)

shows \text{te } p \; (i+1) - \; \text{te } q \; i \leq \text{rmax} + \beta

(proof)

2.2.2 Bounded-drift for logical clocks (IC)

lemma \text{bd}:

assumes \text{ie: } s \leq t

and \text{rb1: } \text{rho-bound1 } C

and \text{rb2: } \text{rho-bound2 } D

and \text{PC-ie: } D \; q \; t - \; D \; q \; s \leq C \; p \; t - \; C \; p \; s

and \text{corr-p: correct } p \; t

and \text{corr-q: correct } q \; t

shows \text{C } p \; t - \; D \; q \; t \mid \leq \mid C \; p \; s - \; D \; q \; s \mid + 2 \ast \rho \ast (t - s)

(proof)

lemma \text{bounded-drift}:

assumes \text{ie: } s \leq t

and \text{rb1: } \text{rho-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| ≤ |C p s - D q s| + 2*q*(t - s)
(proof)

Drift rate of logical clocks

lemma IC-rate1:
rho-bound1 (λ p t. IC p i t)
(proof)

lemma IC-rate2:
rho-bound2 (λ p t. IC p i t)
(proof)

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

definition ICf :: nat ⇒ (process ⇒ time ⇒ Clocktime)
where
ICf i = (λ p t. IC p i t)

lemma IC-bd:
assumes ie: s ≤ t
and corr-p: correct p t
and corr-q: correct q t
shows |IC p i t - IC q j t| ≤ |IC p i s - IC q j s| + 2*q*(t - s)
(proof)

lemma event-bound:
assumes ie1: 0 ≤ (t::real)
and corr-p: correct p t
and corr-q: correct q t
shows ∃ i. t < max (te p i) (te q i)
(proof)

2.3 Agreement property

definition γ1 x = π (2*q*β + 2*Λ) (2*Λ + x + 2*q*(rmax + β))
definition γ2 x = x + 2*q*rmax
definition γ3 x = α (2*Λ + x + 2*q*(rmax + β)) + Λ + 2*q*β

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))| ≤ x)
definition okClocks :: [process, process, nat] ⇒ bool where
okClocks p q i ←→ (∀ t. 0 ≤ t ∧ t < max (te p i) (te q i)
∧ correct p t ∧ correct q t
→ |VC p t - VC q t| ≤ δ)
lemma \textit{okClocks-sym}:
assumes \textit{ok-pq}: \textit{okClocks} p q i
shows \textit{okClocks} q p i
⟨proof⟩

lemma \textit{ICp-Suc}:
assumes \textit{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 \textit{IC-trans-inv}:
assumes \textit{ie1}: te q (i+1) ≤ te p (i+1)
and \textit{corr-p}: correct p (te p (i+1))
and \textit{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))))
⟨proof⟩

lemma \textit{beta-rho}:
assumes \textit{ie}: te q (i+1) ≤ te p (i+1)
and \textit{corr-p}: correct p (te p (i+1))
and \textit{corr-q}: correct q (te p (i+1))
and \textit{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 + 2∗Λ
⟨proof⟩

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

lemma \textit{pe-cond1}:
assumes \textit{ie}: te q (i+1) ≤ te p (i+1)
and \textit{corr-p}: correct p (te p (i+1))
and \textit{corr-q}: correct q (te p (i+1))
and \textit{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
⟨proof⟩

lemma \textit{pe-cond2}:
assumes \textit{ie}: te m i ≤ te l i
and \textit{corr-k}: correct k (te k (i+1))
and \textit{corr-l-tk}: correct l (te k (i+1))
and \textit{corr-m-tk}: correct m (te k (i+1))
and \textit{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 \textit{theta-bound}:
assumes \textit{corr-l}: correct l (te p (i+1))
and \textit{corr-m}: correct m (te p (i+1))
and \textit{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 \[ |ϑ \ p \ (i+1) \ l - ϑ \ p \ (i+1) \ m| \leq 2*Λ + \delta S + 2*rα(max + β) \]
(proof)

**Lemma** four-one-ind-half:
assumes ie1: \( \beta \leq rmin \)
and ie2: \( \mu \leq \delta S \)
and ie3: \( γ1 \ \delta S \leq \delta S \)
and ind-hyp: okmaxsync \( i \ \delta S \)
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 \[ |IC \ p \ (i+1) \ (te \ p \ (i+1)) - IC \ q \ (i+1) \ (te \ p \ (i+1))| \leq \delta S \]
(proof)

Theorem 4.1 in Shankar's paper.

**Theorem** four-one:
assumes ie1: \( \beta \leq rmin \)
and ie2: \( \mu \leq \delta S \)
and ie3: \( γ1 \ \delta S \leq \delta S \)
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 \( p \ (te \ p \ (i+1)) = cfn \ p \ (ϑ \ p \ (i+1)) \)
(proof)

Lemma for the inductive case in Theorem 4.2

**Lemma** four-two-ind:
assumes ie1: \( \beta \leq rmin \)
and ie2: \( \mu \leq \delta S \)
and ie3: \( γ1 \ \delta S \leq \delta S \)
and ie4: \( γ2 \ \delta S \leq \delta \)
and ie5: \( γ3 \ \delta S \leq \delta \)
and ie6: te q \( (i+1) \leq te \ p \ (i+1) \)
and ind-hyp: okClocks \( p \ q \ i \)
and t-bound1: \( 0 \leq t \)
and t-bound2: \( t < max \ (te \ p \ (i+1)) \ (te \ q \ (i+1)) \)
and t-bound3: \( max \ (te \ p \ i) \ (te \ q \ i) \leq t \)
and tνq-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 \[ |VC \ p \ t - VC \ q \ t| \leq \delta \]
(proof)

Theorem 4.2 in Shankar's paper.

**Theorem** four-two:
assumes ie1: \( \beta \leq rmin \)
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: $\beta \leq \text{rmin}$
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: correct p t $\land$ correct q t
shows $|\text{VC} p t - \text{VC} q t| \leq \delta$
(proof)

end

References


