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

Alwen Tiu
April 19, 2020

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
2 Isar proof scripts
   2.1 Types and constants definitions
   2.2 Clock conditions
      2.2.1 Some derived properties of clocks
      2.2.2 Bounded-drift for logical clocks (IC)
   2.3 Agreement property

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

function axiomatication

\( \delta :: \text{real} \) and

\( \mu :: \text{real} \) and

\( \rho :: \text{real} \) and

\( \text{rmin} :: \text{real} \) and

\( \text{rmax} :: \text{real} \) and

\( \beta :: \text{real} \) and

\( \Lambda :: \text{real} \) and

\( \text{np} :: \text{process} \) and

\( \text{maxfaults} :: \text{process} \) and

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

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

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

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

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

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

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

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

**definition**

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

\[ \text{count } f \ n = \text{card } \{ p. p < n \land f p \} \]

**definition**

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

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

**definition**

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

\[ \text{okRead1 } f \ x \ ppred \longleftrightarrow (\forall \ l \ m. \ ppred \ l \land ppred \ m \longrightarrow |f l - f m| \leq x) \]

**definition**

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

\[ \text{okRead2 } f \ g \ x \ ppred \longleftrightarrow (\forall \ p. \ ppred \ p \longrightarrow |f p - g p| \leq x) \]

**definition**

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

\[ \text{rho-bound1 } C \longleftrightarrow (\forall \ p s t. \text{correct } p t \land s \leq t \longrightarrow \text{PC } p t - \text{C } p s \leq (t - s)*(1 + \varrho)) \]

**definition**

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

\[ \text{rho-bound2 } C \longleftrightarrow (\forall \ p s t. \text{correct } p t \land s \leq t \longrightarrow (t - s)*(1 - \varrho) \leq \text{C } p t - \text{C } p s) \]

### 2.2 Clock conditions

Some general assumptions

**axiomatization where**

\[ \text{constants-ax: } 0 < \beta \land 0 < \mu \land 0 < \text{rmin} \]
\[ \land \text{rmin} \leq \text{rmax} \land 0 < \varrho \land 0 < \text{np} \land \text{maxfaults} \leq \text{np} \]

**axiomatization where**

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

**axiomatization where**

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

\[ \text{I_Clock: } \forall \, p \, t \, i. \, \text{correct} \, p \, t \rightarrow \text{I_C} \, p \, i \, t = PC \, p \, t + \text{Adj} \, p \, i \]

Condition 1: initial skew

axiomatization where

\[ \text{init: } \forall \, p. \, \text{correct} \, p \, 0 \rightarrow 0 \leq PC \, p \, 0 \land PC \, p \, 0 \leq \mu \]

Condition 2: bounded drift

axiomatization where

\[ \text{rate-1: } \forall \, p \, s \, t. \, \text{correct} \, p \, t \land s \leq t \rightarrow PC \, p \, t - PC \, p \, s \leq (t - s) \ast (1 + \varrho) \quad \text{and} \]
\[ \text{rate-2: } \forall \, p \, s \, t. \, \text{correct} \, p \, t \land s \leq t \rightarrow (t - s) \ast (1 - \varrho) \leq PC \, p \, t - PC \, p \, s \]

Condition 3: bounded interval

axiomatization where

\[ \text{rts0: } \forall \, p \, t \, i. \, \text{correct} \, p \, t \land te \, p \, (i + 1) \rightarrow t - te \, p \, i \leq \text{rmax} \quad \text{and} \]
\[ \text{rts1: } \forall \, p \, t \, i. \, \text{correct} \, p \, t \land te \, p \, (i + 1) \leq t \rightarrow \text{rmin} \leq t - te \, p \, i \]

Condition 4: bounded delay

axiomatization where

\[ \text{rts2a: } \forall \, p \, q \, t \, i. \, \text{correct} \, p \, t \land \text{correct} \, q \, t \land te \, q \, i + \beta \leq t \rightarrow te \, p \, i \leq t \quad \text{and} \]
\[ \text{rts2b: } \forall \, p \, q \, i. \, \text{correct} \, p \, (te \, p \, i) \land \text{correct} \, q \, (te \, q \, i) \rightarrow \text{abs}((te \, p \, i - te \, q \, i) \leq \beta \]

Condition 5: initial synchronization

axiomatization where

\[ \text{synch0: } \forall \, p. \, te \, p \, 0 = 0 \]

Condition 6: nonoverlap

axiomatization where

\[ \text{nonoverlap: } \beta \leq \text{rmin} \]

Condition 7: reading errors

axiomatization where

\[ \text{readererror: } \forall \, p \, q \, i. \, \text{correct} \, p \, (te \, p \, (i + 1)) \land \text{correct} \, q \, (te \, p \, (i + 1)) \rightarrow \]
\[ \text{abs}(\varrho \, p \, (i + 1) \, q - IC \, q \, i \, (te \, p \, (i + 1))) \leq \Lambda \]

Condition 8: bounded faults

axiomatization where

\[ \text{correct-closed: } \forall \, p \, s \, t. \, s \leq t \land \text{correct} \, p \, t \rightarrow \text{correct} \, p \, s \quad \text{and} \]
\[ \text{correct-count: } \forall \, t. \, np - \text{maxfaults} \leq \text{count} \, (\lambda \, p. \, \text{correct} \, p \, t) \, np \]

Condition 9: Translation invariance

axiomatization where

\[ \text{trans-inv: } \forall \, p \, f \, x. \, 0 \leq x \rightarrow cfn \, p \, (\lambda \, y. \, f \, y + x) = cfn \, p \, f \, x \]

Condition 10: precision enhancement

axiomatization where

\[ \text{prec-ehn: } \forall \, ppred \, p \, q \, f \, g \, x \, y. \]
\[ \text{np} - \text{maxfaults} \leq \text{count} \, ppred \, np \land \]
\[ \text{okRead1} \, f \, y \, ppred \land \text{okRead1} \, g \, y \, ppred \land \]

\[ 4 \]
\[
\text{okRead} 2 \ f \ g \ x \ \text{ppred} \land \ \text{ppred} \ p \land \ \text{ppred} \ q \\
\rightarrow \ \text{abs}(cfn \ p \ f - cfn \ q \ g) \leq \pi \ x \ y
\]

Condition 11: accuracy preservation

axiomatization where

\[
\forall \ \text{ppred} \ p \ q \ f \ x. \ \text{okRead} 1 \ f \ x \ \text{ppred} \land \ \text{maxfaults} \leq \ \text{count} \ \text{ppred} \ np \\
\land \ \text{ppred} \ p \land \ \text{ppred} \ q \rightarrow \ \text{abs}(cfn \ p \ f - cfn \ q \ g) \leq \alpha \ x
\]

2.2.1 Some derived properties of clocks

\[\text{lemma rts0d:} \]
assumes \(cp: \text{correct} \ (te \ p \ (i+1))\)
shows \(te \ p \ (i+1) - te \ p \ i \leq \ rmax\)
\(\langle \text{proof} \rangle\)

\[\text{lemma rts1d:} \]
assumes \(cp: \text{correct} \ (te \ p \ (i+1))\)
shows \(rmin \leq te \ p \ (i+1) - te \ p \ i\)
\(\langle \text{proof} \rangle\)

\[\text{lemma rte:} \]
assumes \(cp: \text{correct} \ (te \ p \ (i+1))\)
shows \(te \ p \ i \leq te \ p \ (i+1)\)
\(\langle \text{proof} \rangle\)

\[\text{lemma beta-bound1:} \]
assumes \(\text{corr-p: correct} \ (te \ p \ (i+1))\)
and \(\text{corr-q: correct} \ (te \ q \ i)\)
shows \(0 \leq te \ p \ (i+1) - te \ q \ i\)
\(\langle \text{proof} \rangle\)

\[\text{lemma beta-bound2:} \]
assumes \(\text{corr-p: correct} \ (te \ p \ (i+1))\)
and \(\text{corr-q: correct} \ (te \ q \ i)\)
shows \(te \ p \ (i+1) - te \ q \ i \leq \ rmax + \beta\)
\(\langle \text{proof} \rangle\)

2.2.2 Bounded-drift for logical clocks (IC)

\[\text{lemma bd:} \]
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 \(corr-p: \text{correct} \ p \ t\)
and \(corr-q: \text{correct} \ q \ t\)
shows \(|C \ p \ t - D \ q \ t| \leq |C \ p \ s - D \ q \ s| + 2*\rho*(t - s)\)
\(\langle \text{proof} \rangle\)

\[\text{lemma bounded-drift:} \]
assumes \(ie: s \leq t\)
and \(rb1: \ \rho-bound1 \ C\)
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| \leq |C p s - D q s| + 2*\(\rho\)*\((t - s)\)\)

(\textit{proof})

Drift rate of logical clocks

\textbf{lemma IC-rate1:}
\[\text{rho-bound1} \ (\lambda \ p \ t. \ IC \ p \ i \ t)\]

(\textit{proof})

\textbf{lemma IC-rate2:}
\[\text{rho-bound2} \ (\lambda \ p \ t. \ IC \ p \ i \ t)\]

(\textit{proof})

Auxiliary 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 \(corr-p\): 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*\(\rho\)*\((t - s)\)\)

(\textit{proof})

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

(\textit{proof})

\textbf{2.3 Agreement property}

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

\textbf{definition} \(\gamma 2 \ x = x + 2*\(\rho\)*rmax\)

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

\textbf{definition}
\[\text{okmaxsync} :: [nat, \text{Clocktime}] \Rightarrow \text{bool}\]

\textbf{where}
\[\text{okmaxsync} \ i \ x \longleftrightarrow (\forall \ p \ q. \ \text{correct} \ p \ \text{max} \ (te \ p \ i) \ (te \ q \ i))\]
\[
\land \ \text{correct} \ q \ \text{max} \ (te \ p \ i) \ (te \ q \ i) \longrightarrow
|IC \ p \ i \ \text{max} \ (te \ p \ i) \ (te \ q \ i)) - IC \ q \ i \ \text{max} \ (te \ p \ i) \ (te \ q \ i))| \leq x
\]

\[\text{definition}
\[\text{okClocks} :: [\text{process}, \text{process}, \text{nat}] \Rightarrow \text{bool}\]

\textbf{where}
\[\text{okClocks} \ p \ q \ i \longleftrightarrow (\forall \ t. \ 0 \leq t \land t < \text{max} \ (te \ p \ i) \ (te \ q \ i))\]
\[
\land \ \text{correct} \ p \ t \land \text{correct} \ q \ t
\longrightarrow |VC \ p \ t - VC \ q \ t| \leq \delta
\]
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)

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*Δ + 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:
\[ |IC l i (max (te l i) (te m i)) − IC m i (max (te l i) (te m i))| \leq δS \]
sows \[ |θ p (i+1) l − θ p (i+1) m| \leq 2Λ + δS + 2μ(ρmax + β) \]
\( ⟨proof⟩ \)

\textbf{Lemma four-one-ind-half:}
\begin{itemize}
  \item assumes ie1: \( β \leq rmin \)
  \item and ie2: \( μ ≤ δS \)
  \item and ie3: \( γ1 δS ≤ δS \)
  \item and ind-hyp: okmaxsync i δS
  \item and corr-p: correct p (te p (i+1))
  \item and corr-q: correct q (te p (i+1))
\end{itemize}
sows \[ |IC p (i+1) (te p (i+1)) − IC q (i+1) (te p (i+1))| \leq δS \]
\( ⟨proof⟩ \)

\textbf{Theorem 4.1 in Shankar’s paper.}
\textbf{Theorem four-one:}
\begin{itemize}
  \item assumes ie1: \( β \leq rmin \)
  \item and ie2: \( μ ≤ δS \)
  \item and ie3: \( γ1 δS ≤ δS \)
\end{itemize}
sows okmaxsync i δS
\( ⟨proof⟩ \)

\textbf{Lemma for the inductive case in Theorem 4.2}
\textbf{Lemma four-two-ind:}
\begin{itemize}
  \item assumes ie1: \( β ≤ rmin \)
  \item and ie2: \( μ ≤ δS \)
  \item and ie3: \( γ1 δS ≤ δS \)
  \item and ie4: \( γ2 δS ≤ δ \)
  \item and ie5: \( γ3 δS ≤ δ \)
  \item and ie6: \( te q (i+1) ≤ te p (i+1) \)
  \item and ind-hyp: okClocks p q i
  \item and t-bound1: \( 0 ≤ t \)
  \item and t-bound2: \( t < max (te p (i+1)) (te q (i+1)) \)
  \item and t-bound3: \( max (te p i) (te q i) ≤ t \)
  \item and tqp-bound: \( max (te p i) (te q i) < max (te p (i+1)) (te q (i+1)) \)
  \item and corr-p: correct p t
  \item and corr-q: correct q t
\end{itemize}
sows \[ |VC p t − VC q t| ≤ δ \]
\( ⟨proof⟩ \)

\textbf{Theorem 4.2 in Shankar’s paper.}
\textbf{Theorem four-two:}
\begin{itemize}
  \item assumes ie1: \( β ≤ rmin \)
\end{itemize}
\begin{proof}
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
\end{proof}

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

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

\end{proof}

\section*{References}


