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

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

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

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

\( r_{min} :: \text{real and} \)

\( r_{max} :: \text{real and} \)

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

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

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

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

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

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

\( te :: [\text{process, event}] \rightarrow \text{time and} \)
θ :: [process, event] ⇒ (process ⇒ Clocktime) and

IC :: [process, event, time] ⇒ Clocktime and

correct :: [process, time] ⇒ bool and

cfn :: [process, (process ⇒ Clocktime)] ⇒ Clocktime and

π :: [Clocktime, Clocktime] ⇒ Clocktime and

α :: Clocktime ⇒ Clocktime

definition

count :: [process ⇒ bool, process] ⇒ nat where

count f n = card \{ p. p < n ∧ f p\}

definition

Adj :: [process, event] ⇒ Clocktime where

Adj = (λ p i. if 0 < i then cfn p (θ p i) − PC p (te p i) else 0)

definition

okRead1 :: [process ⇒ Clocktime, Clocktime, process ⇒ bool] ⇒ bool where

okRead1 f x ppred ←→ (∀ l m. ppred l ∧ ppred m → |f l − f m| ≤ x)

definition

okRead2 :: [process ⇒ Clocktime, process ⇒ Clocktime, Clocktime,

process ⇒ bool] ⇒ bool where

okRead2 f g x ppred ←→ (∀ p. ppred p → |f p − g p| ≤ x)

definition

rho-bound1 :: [[process, time] ⇒ Clocktime] ⇒ bool where

rho-bound1 C ←→ (∀ p s t. correct p t ∧ s ≤ t → C p t − C p s ≤ (t − s)∗(1 + ϱ))

definition

rho-bound2 :: [[process, time] ⇒ Clocktime] ⇒ bool where

rho-bound2 C ←→ (∀ p s t. correct p t ∧ s ≤ t → (t − s)∗(1 − ϱ) ≤ C p t − C p s)

2.2 Clock conditions

Some general assumptions

axiomatization where

constants-ax: 0 < β ∧ 0 < μ ∧ 0 < rmin

∧ rmin ≤ rmax ∧ 0 < ϱ ∧ 0 < np ∧ maxfaults ≤ np

axiomatization where

PC-monotone: ∀ p s t. correct p t ∧ s ≤ t → PC p s ≤ PC p t

axiomatization where

VClock: ∀ p t i. correct p t ∧ te p i ≤ t ∧ t < te p (i + 1) → VC p t = IC p i t
axiomatization where
IClock: $\forall \ p \ t \ i. \ correct \ p \ t \rightarrow IC \ p \ i = PC \ p \ t + Adj \ p \ i$

Condition 1: initial skew

axiomatization where
init: $\forall \ p. \ correct \ p \ 0 \rightarrow 0 \leq PC \ p \ 0 \land PC \ p \ 0 \leq \mu$

Condition 2: bounded drift

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

Condition 3: bounded interval

axiomatization where
$\forall \ p \ t \ i. \ correct \ p \ t \wedge t \leq te \ p \ (i+1) \rightarrow t - te \ p \ i \leq rmax$ and
$\forall \ p \ t \ i. \ correct \ p \ t \wedge te \ p \ (i+1) \leq t \rightarrow rmin \leq t - te \ p \ i$

Condition 4: bounded delay

axiomatization where
$\forall \ p \ q \ t \ i. \ correct \ p \ t \wedge q \ t \wedge te \ q \ i + \beta \leq t \rightarrow te \ p \ i \leq t$ and
$\forall \ p \ q \ t \ i. \ correct \ p \ te \ p \ i \wedge correct \ q \ (te \ q \ i) \rightarrow abs(\ te \ p \ i - te \ q \ i) \leq \beta$

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

Condition 8: bounded faults

axiomatization where
$\forall \ p \ s \ t. \ s \leq t \rightarrow correct \ p \ t \rightarrow correct \ p \ s$ and
$\forall \ t. \ np - maxfaults \leq count \ (\lambda \ p. \ correct \ p \ t) \ np$

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: $\forall \ ppred \ p \ q \ g \ x \ y.
np - maxfaults \leq count \ ppred \ np \wedge
okRead1 \ f \ y \ ppred \wedge okRead1 \ g \ y \ ppred \wedge$
\begin{align*}
  \text{okRead2 } & \text{f } g \text{ x ppred } \land \text{ppred } \land \text{ppred } q \\
  \rightarrow & \text{abs(}c\text{fn } f \text{ - } c\text{fn } q \text{ g}) \leq \pi \text{ x y}
\end{align*}

Condition 11: accuracy preservation

**axiomatization where**

\text{acc-prsv:}

\forall \text{ ppred } p \text{ q f x}. \text{ okRead1 } f \text{ x ppred } \land \text{ np - maxfaults } \leq \text{ count ppred np } \\
\land \text{ ppred } p \land \text{ ppred } q \rightarrow \text{abs(}c\text{fn } f \text{ - } f \text{ q}) \leq \alpha \text{ x}

\[2.2.1\] Some derived properties of clocks

**lemma** \text{rts0d:}

**assumes** \text{cp: correct } (te \text{ p (i+1))} \\
**shows** \text{te p (i+1) - te p i} \leq \text{rmax}

**proof**

**lemma** \text{rts1d:}

**assumes** \text{cp: correct } (te \text{ p (i+1))} \\
**shows** \text{rmin} \leq \text{te p (i+1) - te p i}

**proof**

**lemma** \text{rte:}

**assumes** \text{cp: correct } (te \text{ p (i+1))} \\
**shows** \text{te p i} \leq \text{te p (i+1)}

**proof**

**lemma** \text{beta-bound1:}

**assumes** \text{corr-p: correct } (te \text{ p (i+1))} \\
and \text{corr-q: correct } (te \text{ q i}) \\
**shows** \text{0} \leq \text{te p (i+1) - te q i}

**proof**

**lemma** \text{beta-bound2:}

**assumes** \text{corr-p: correct } (te \text{ p (i+1))} \\
and \text{corr-q: correct } (te \text{ q i}) \\
**shows** \text{te p (i+1) - 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 \text{t} \\
and \text{rb1: } \text{rho-bound1 C} \\
and \text{rb2: } \text{rho-bound2 D} \\
and \text{PC-ie: } D \text{ q t} - D \text{ q s} \leq C \text{ p t} - C \text{ p s} \\
and \text{corr-p: correct } \text{p t} \\
and \text{corr-q: correct } \text{q t} \\
**shows** \text{| } C \text{ p t} - D \text{ q t} \text{|} \leq \text{| } C \text{ p s} - D \text{ q s} \text{|} + 2\text{g*}(t - s)

**proof**

**lemma** \text{bounded-drift:}

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

**Drift rate of logical clocks**

**Lemma** IC-rate1:
\[\rho-bound1 (\lambda p t. IC p \ i \ t)\]

**Lemma** IC-rate2:
\[\rho-bound2 (\lambda p t. IC p \ i \ t)\]

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

**Definition**
\[ICf :: nat \Rightarrow (process \Rightarrow time \Rightarrow \text{Clocktime})\]

\[ICf \ i = (\lambda p t. IC p \ i \ t)\]

**Lemma** IC-bd:
\[\text{assumes ie: } s \leq t\]
\[\text{and corr-p: correct p t}\]
\[\text{and corr-q: correct q t}\]
\[\text{shows } |IC p \ i \ t - IC q j \ t| \leq |IC p \ i \ s - IC q j \ s| + 2 \cdot g^*(t - s)\]

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

### 2.3 Agreement property

**Definition** \[\gamma 1 x = \pi (2 \cdot g^* \beta + 2 \cdot \Lambda) (2 \cdot \Lambda + x + 2 \cdot g^*(\mathrm{rmax} + \beta))\]

**Definition** \[\gamma 2 x = x + 2 \cdot g^* \mathrm{rmax}\]

**Definition** \[\gamma 3 x = \alpha (2 \cdot \Lambda + x + 2 \cdot g^*(\mathrm{rmax} + \beta)) + \Lambda + 2 \cdot g^* \beta\]

**Definition**
\[\text{okmaxsync :: } [\text{nat, Clocktime}] \Rightarrow \text{bool where}\]
\[\text{okmaxsync \ i \ x} \longleftrightarrow (\forall p q. \text{correct p} (\max (te p \ i ) (te q \ i))\]
\[\text{\& correct q} (\max (te p \ i ) (te q \ i)) \longrightarrow |IC p \ i \ (\max (te p \ i ) (te q \ i)) - IC q \ i \ (\max (te p \ i ) (te q \ i))| \leq x\]

**Definition**
\[\text{okClocks :: } [\text{process, process, nat}] \Rightarrow \text{bool where}\]
\[\text{okClocks} \ p \ q \ i \longleftrightarrow (\forall t. 0 \leq t \land t < \max (te p \ i ) (te q \ i))\]
\[\text{\& correct p \ t \land correct q \ t} \]
\[\longrightarrow |VC p \ t - VC q \ t| \leq \delta\]
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-lk: correct l (te k (i+1))
and corr-mdk: 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:

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

shows \(|\partial p \ (i+1) \ l - \partial p \ (i+1) \ m| \leq 2*\Lambda + \delta S + 2*\varrho(rmax + \beta)
\]

(\text{proof})

\textbf{lemma four-one-ind-half:}

\textbf{assumes ie1: } \beta \leq rmin
\textbf{and ie2: } \mu \leq \delta S
\textbf{and ie3: } \gamma 1 \ \delta S \leq \delta S
\textbf{and ind-hyp: okmaxsync i } \delta S
\textbf{and corr-p: correct p (te p (i+1))}
\textbf{and corr-q: correct q (te p (i+1))}

\textbf{shows } |IC p \ (i+1) \ (te p \ (i+1)) - IC q \ (i+1) \ (te p \ (i+1))| \leq \delta S
\]

(\text{proof})

\textbf{Theorem 4.1 in Shankar’s paper.}

\textbf{theorem four-one:}

\textbf{assumes ie1: } \beta \leq rmin
\textbf{and ie2: } \mu \leq \delta S
\textbf{and ie3: } \gamma 1 \ \delta S \leq \delta S
\textbf{shows okmaxsync i } \delta S
\textbf{(proof)}

\textbf{lemma VC-cfn:}

\textbf{assumes corr-p: correct p (te p (i+1))}
\textbf{and ie: } te p \ (i+1) < te p \ (i+2)
\textbf{shows VC p (te p (i+1)) = cfn p (\partial p (i+1))}
\textbf{(proof)}

\textbf{Lemma for the inductive case in Theorem 4.2}

\textbf{lemma four-two-ind:}

\textbf{assumes ie1: } \beta \leq rmin
\textbf{and ie2: } \mu \leq \delta S
\textbf{and ie3: } \gamma 1 \ \delta S \leq \delta S
\textbf{and ie4: } \delta S \leq \delta
\textbf{and ie5: } \gamma 3 \ \delta S \leq \delta
\textbf{and ie6: } te q \ (i+1) \leq te p \ (i+1)
\textbf{and ind-hyp: okClocks p q i}
\textbf{and t-bound1: } 0 \leq t
\textbf{and t-bound2: } t < \max (te p \ (i+1)) \ (te q \ (i+1))
\textbf{and t-bound3: } \max (te p \ i) \ (te q \ i) \leq t
\textbf{and tsp-bound: } \max (te p \ i) \ (te q \ i) < \max (te p \ (i+1)) \ (te q \ (i+1))
\textbf{and corr-p: correct p t}
\textbf{and corr-q: correct q t}
\textbf{shows } |VC p t - VC q t| \leq \delta
\textbf{(proof)}

\textbf{Theorem 4.2 in Shankar’s paper.}

\textbf{theorem four-two:}

\textbf{assumes ie1: } \beta \leq rmin
and \( i e2: \mu \leq \delta S \)
and \( i e3: \gamma 1 \delta S \leq \delta S \)
and \( i e4: \gamma 2 \delta S \leq \delta \)
and \( i e5: \gamma 3 \delta S \leq \delta \)
shows okClocks p q i
(proof)

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

**Theorem agreement:**
- assumes \( i e1: \beta \leq rmin \)
and \( i e2: \mu \leq \delta S \)
and \( i e3: \gamma 1 \delta S \leq \delta S \)
and \( i e4: \gamma 2 \delta S \leq \delta \)
and \( i e5: \gamma 3 \delta S \leq \delta \)
and \( i e6: 0 \leq t \)
and \( cpq: \text{correct } p t \wedge \text{correct } q t \)
shows \(|VC p t - VC q t| \leq \delta \)
(proof)

end

References


