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

Alwen Tiu

June 11, 2019

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) ........................................... 6
   2.3 Agreement property ............................................................... 10

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.

\textbf{type-synonym} \textit{process} = \textit{nat}

\textbf{type-synonym} \textit{event} = \textit{nat}

\textbf{type-synonym} \textit{time} = \textit{real}

\textbf{type-synonym} \textit{Clocktime} = \textit{real}

\textbf{axiomatization}

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

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

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

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

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

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

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

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

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

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

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

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

2
\[ \emptyset :: \text{[process, event]} \Rightarrow \text{(process } \Rightarrow \text{ Clocktime)} \text{ and} \]

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

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

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

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

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

\[ \text{definition} \]

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

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

\[ \text{definition} \]

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

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

\[ \text{definition} \]

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

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

\[ \text{definition} \]

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

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

\[ \text{definition} \]

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

\[ \text{rho-bound1 } C \leftrightarrow (\forall p s t. \text{correct } p t \land s \leq t \Longrightarrow C p t - C p s \leq (t-s)*(1 + \rho)) \]

\[ \text{definition} \]

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

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

### 2.2 Clock conditions

Some general assumptions

**axiomatization where**

- constants-\text{-ax}: \(0 < \beta \land 0 < \mu \land 0 < \text{rmin}\)
- \(\text{rmin} \leq \text{rmax} \land 0 < \rho \land 0 < \text{np} \land \text{maxfaults} \leq \text{np}\)

**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

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 + s) and
rate-2: ∀p s t. correct p t ∧ s ≤ t → (t - s) > (1 - s) ≤ 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 p i + β ≤ t → te p i ≤ t and
rts2b: ∀p q i. correct p (te p i) ∧ correct q (te p i) → abs((te p i - te p 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 (λp. correct p t) np

Condition 9: Translation invariance

axiomatization where

trans-inv: ∀p f x. 0 ≤ x → cfn p (λy. 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 ∧
\[
okRead2 \ f \ g \ x \ ppred \land ppred \land ppred \ q \\
\rightarrow \ \text{abs}(\text{cfn} \ p \ f - \text{cfn} \ q \ g) \leq \pi \ x \ y
\]

Condition 11: accuracy preservation

axiomatization where

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

### 2.2.1 Some derived properties of clocks

**Lemma rts0d:**

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

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

using \( cp \ \text{rts0} \) by simp

**Lemma rts1d:**

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

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

using \( cp \ \text{rts1} \) by simp

**Lemma rte:**

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

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

proof–

from \( cp \ \text{rts1d} \) have \( \text{rmin} \leq \text{te} \ p \ (i+1) - \text{te} \ p \ i \)

by simp

from this \( \text{constants-ax} \) show \( \? \text{thesis} \) by arith

qed

**Lemma beta-bound1:**

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

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

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

proof–

from \( \text{corr-p} \ \text{rte} \) have \( \text{te} \ p \ i \leq \text{te} \ p \ (i+1) \)

by simp

from this \( \text{corr-p} \ \text{correct-closed} \) have \( \text{corr-pi}: \text{correct} \ p \ (\text{te} \ p \ i) \)

by blast

from \( \text{corr-p} \ \text{rts1d} \ \text{nonoverlap} \) have \( \text{rmin} \leq \text{te} \ p \ (i+1) - \text{te} \ p \ i \)

by simp

from this \( \text{nonoverlap} \) have \( \beta \leq \text{te} \ p \ (i+1) - \text{te} \ p \ i \) by simp

hence \( \text{te} \ p \ i + \beta \leq \text{te} \ p \ (i+1) \) by simp

from this \( \text{corr-p} \ \text{corr-q} \ \text{rts2a} \)

have \( \text{te} \ q \ i \leq \text{te} \ p \ (i+1) \)

by blast

thus \( \? \text{thesis} \) by simp

qed

**Lemma beta-bound2:**

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

and \( \text{corr-q}: \text{correct} \ q \ (\text{te} \ q \ i) \)
shows $te p (i+1) - te q i \leq r_{\text{max}} + \beta$

proof –
from corr-p rte have $te p i \leq te p (i+1)$
  by simp
from this corr-p correct-closed have corr-pi: correct $p (te p i)$
  by blast
have split: $te p (i+1) - te q i =$
  $(te p (i+1) - te p i) + (te p i - te q i)$
  by (simp)
from corr-q corr-pi rts2b have Eq1: $\text{abs}(te p i - te q i) \leq \beta$
  by simp
have Eq2: $te p i - te q i \leq \beta$
proof cases
  assume $te q i \leq te p i$
  from this Eq1 show ?thesis
    by (simp add: abs-if)
next
  assume $\neg (te q i \leq te p i)$
  from this Eq1 show ?thesis
    by (simp add: abs-if)
qed
from corr-p rts0d have $te p (i+1) - te p i \leq r_{\text{max}}$
  by simp
from this split Eq2 show ?thesis by simp
qed

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 $| C p t - D q t | \leq | C p s - D q s | + 2*q*(t - s)$
proof –
let $?Dt = C p t - D q t$
let $?Ds = C p s - D q s$
let $?Bp = C p t - C p s$
let $?Bq = D q t - D q s$
let $?I = t - s$

have $| ?Bp - ?Bq | \leq 2*q*(t - s)$
proof –
from PC-ie have Eq1: $| ?Bp - ?Bq | = ?Bp - ?Bq$ by (simp add: abs-if)
from corr-p ie rb1 have Eq2: $?Bp - ?Bq \leq ?I*(1+q) - ?Bq$ (is $?E1 \leq ?E2$)
  by(simp add: rho-bound1-def)
from corr-q ie rb2 have $?I*(1 - q) \leq ?Bq$
  by(simp add: rho-bound2-def)
from this have Eq3: \( E_2 \leq I^*(1+\varrho) - I^*(1-\varrho) \)

by (simp)

have Eq4: \( I^*(1+\varrho) - I^*(1-\varrho) = 2*\varrho*I \)

by (simp add: algebra-simps)

from Eq1 Eq2 Eq3 Eq4 show thesis by simp

qed

moreover have \( |D_t| \leq |B_p - B_q| + |D_s| \)

by (simp add: abs-if)

ultimately show thesis by simp

qed

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 \( |C_p t - D_q t| \leq |C_p s - D_q s| + 2*\varrho*(t-s) \)

proof

let \( B_p = C_p t - C_p s \)

let \( B_q = D_q t - D_q s \)

show thesis

proof cases

assume \( B_q \leq B_p \)

from this ie rb1 rb4 corr-p corr-q bd show thesis by simp

next

assume \( \neg (B_q \leq B_p) \)

hence \( B_p \leq B_q \) by simp

from this ie rb2 rb3 corr-p corr-q bd

have \( |D_q t - C_p t| \leq |D_q s - C_p s| + 2*\varrho*(t-s) \)

by simp

from this show thesis by (simp add: abs-minus-commute)

qed

qed

Drift rate of logical clocks

lemma IC-rate1:

rho-bound1 \( (\lambda p t. IC_p i t) \)

proof

fix \( p::process \)

fix \( s::time \)

fix \( t::time \)

assume \( cp: correct p t \)

assume \( ie: s \leq t \)

from \( cp ie \) correct-closed have \( cps: correct p s \)

by blast

have \( IC_p i t - IC_p i s \leq (t-s)*(1+\varrho) \)

proof

Drift rate of logical clocks
from \( cp\) IClock have \( IC\ p\ i\ t = PC\ p\ t + Adj\ p\ i\)
  by simp
moreover
from \( cps\) IClock have \( IC\ p\ i\ s = PC\ p\ s + Adj\ p\ i\)
  by simp
moreover
from \( cp\) ie rate-1 have \( PC\ p\ t - PC\ p\ s \leq (t - s)\cdot (1 + \varrho)\)
  by simp
ultimately show \(?thesis\) by simp
qed

thus \(?thesis\) by (simp add: rho-bound1-def)
qed

lemma IC-rate2:
  rho-bound2 \((\lambda p\ t.\ IC\ p\ i\ t)\)
proof
  {  
    fix \( p::process\)
    fix \( s::time\)
    fix \( t::time\)
    assume \( cp:: correct\ p\ t\)
    assume \( ie:: s \leq t\)
    from \( cp\) ie correct-closed have \( cps:: correct\ p\ s\)
      by blast
    have \((t - s) \cdot (1 - \varrho) \leq IC\ p\ i\ t - IC\ p\ i\ s\)
      proof
        from \( cp\) IClock have \( IC\ p\ i\ t = PC\ p\ t + Adj\ p\ i\)
          by simp
        moreover
        from \( cps\) IClock have \( IC\ p\ i\ s = PC\ p\ s + Adj\ p\ i\)
          by simp
        moreover
        from \( cp\) ie rate-2 have \((t - s) \cdot (1 - \varrho) \leq PC\ p\ t - PC\ p\ s\)
          by simp
        ultimately show \(?thesis\) by simp
      qed
  }
  thus \(?thesis\) by (simp add: rho-bound2-def)
qed

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

definition
  \( ICf :: nat \Rightarrow (process \Rightarrow time \Rightarrow Clocktime)\) 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 \(| IC\ p\ i\ t - IC\ q\ j\ t| \leq | IC\ p\ i\ s - IC\ q\ j\ s| + 2 \cdot g\cdot (t - s)\)
proof
  let \( ?C = ICf\ i\)
let $D = I(Cf j)$

let $G = |C p t - D q t| \leq |C p s - D q s| + 2 \times q \times (t - s)$

from `IC-rate1` have `rb1`: `rho-bound1` $(ICf i)$ $\land$ `rho-bound1` $(ICf j)$
  by `(simp add: ICf-def)`

from `IC-rate2` have `rb2`: `rho-bound2` $(ICf i)$ $\land$ `rho-bound2` $(ICf j)$
  by `(simp add: ICf-def)`

from `ie rb1 rb2 corr-p corr-q bounded-drift`
have $G$ by `(simp add: ICf-def)`

from this show `thesis` by `(simp add: ICf-def)`
qed

lemma `event-bound`:
assumes `ie1`: $0 \leq (t :: \text{real})$
and `corr-p`: `correct p t`
and `corr-q`: `correct q t`
shows $\exists \ i. \ t < \text{max} (te p i) (te q i)$
proof `(rule ccontr)`
  assume `A`: $\neg (\exists \ i. \ t < \text{max} (te p i) (te q i))$
  show `False`
  proof
  have `F1`: $\forall \ i. \ te p i \leq t$
    proof
      fix $i :: \text{natic}$
      from `A` have $\neg (t < \text{max} (te p i) (te q i))$
        by `(simp)`
      hence `Eq1`: $\text{max} (te p i) (te q i) \leq t$ `by arith`
      have `Eq2`: $te p i \leq \text{max} (te p i) (te q i)$
        `by `(simp add: max-def)`
      from `Eq1 Eq2` show $te p i \leq t$ by `(simp)`
    qed
  have `F2`: $\forall (i :: \text{natic}). \ correct p (te p i)$
    proof
      fix $i :: \text{natic}$
      from `F1` have $te p i \leq t$ `by simp`
      from this `corr-p correct-closed`
      show `correct p (te p i)` `by blast`
    qed
  have `F3`: $\forall (i :: \text{natic}). \ \text{real} i \times rmin \leq te p i$
    proof
      fix $i :: \text{natic}$
      show $\text{real} i \times rmin \leq te p i$
        proof `(induct i)`
        from `synch0` show $(0 :: \text{natic}) \times rmin \leq te p 0$ `by simp`
        next
        fix $i :: \text{natic}$
        assume `ind-hyp`: $\text{real} i \times rmin \leq te p i$
        show $\text{real} (Suc i) \times rmin \leq te p (Suc i)$
proof –

have Eq1: real i * rmin + rmin = (real i + 1)*rmin
  by (simp add: distrib-right)
have Eq2: real i + 1 = real (i+1) by simp
from Eq1 Eq2
have Eq3: real i * rmin + rmin = real (i+1) * rmin
  by presburger
from F2 have cp1: correct p (te p (i+1))
  by simp
from F2 have cp2: correct p (te p i)
  by simp
from cp1 rts1d have rmin ≤ te p (i+1) - te p i
  by simp
hence Eq4: te p i + rmin ≤ te p (i+1) by simp
from ind-hyp have real i * rmin + rmin ≤ te p i + rmin
  by (simp)
from this Eq4 have real i * rmin + rmin ≤ te p (i+1)
  by simp
from this Eq3 show ?thesis by simp
qed
qed
qed

have F4: \( \forall (i::nat). \text{real } i \times rmin \leq t \)
proof
  fix i::nat
  from F1 have te p i \leq t by simp
  moreover
  from F3 have real i * rmin \leq te p i by simp
  ultimately show real i * rmin \leq t by simp
qed
from constants-ax have 0 < rmin by simp
from this reals-Archimedean3
have Archi: \( \exists (k::nat). t < \text{real } k \times rmin \)
  by blast
from Archi obtain k::nat where C: t < real k * rmin ..
from F4 have real k * rmin \leq t by simp
hence notC: \( t < \text{real } k \times rmin \) by simp
from C notC show False by simp
qed
qed

2.3 Agreement property

definition \( \gamma_1 x = \pi (2*\varrho*\beta + 2*\Lambda) (2*\Lambda + x + 2*\varrho*(rmax + \beta)) \)
definition \( \gamma_2 x = x + 2*\varrho*rmax \)
\[ \gamma x = \alpha (2 \Lambda + x + 2 \gamma(r_{\max} + \beta)) + \Lambda + 2 \gamma \beta \]

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

definition\ okClocks \::\ [\text{process}, \text{process}, \text{nat}] \Rightarrow \text{bool} \text{ where}
\[ \text{okClocks}\ p\ q\ i \leftrightarrow (\forall\ t.\ 0 \leq t \land t < \max\ (\text{te}\ q\ i)\ (\text{te}\ p\ i)) \]
\[ \land\ \text{correct}\ p\ t \land\ \text{correct}\ q\ t \rightarrow |VC\ p\ t - VC\ q\ t| \leq \delta \]

lemma okClocks-sym;
assumes ok-pq: okClocks p q i
shows okClocks q p i
proof
\[
\begin{align*}
\text{fix } t &: \text{time} \\
\text{assume } \text{ie1}: 0 &\leq t \\
\text{assume } \text{ie2}: t &< \max(\text{te} q i) (\text{te} p i) \\
\text{assume } \text{corr-q}: \text{correct} q t \\
\text{assume } \text{corr-p}: \text{correct} p t \\
\text{have } \max(\text{te} q i) (\text{te} p i) &= \max(\text{te} p i) (\text{te} q i) \\
&\quad \text{by (simp add: max-def)} \\
&\quad \text{from this ok-pq ie1 ie2 corr-p corr-q} \\
\text{have } |VC q t - VC p t| &\leq \delta \\
&\quad \text{by (simp add: abs-minus-commute okClocks-def)} \\
\end{align*}
\]
thus ?thesis by (simp add: okClocks-def)
qed

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))
using corr-p Iclock by (simp add: Adj-def)

lemma IC-trans-inv:
assumes ie1: 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 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))))
\text{(is } ?T1 = ?T2) 
proof
\[
\begin{align*}
\text{let } ?X &= PC q (te p (i+1)) - PC q (te q (i+1)) \\
&\quad \text{by (simp add: le-diff-eq)} \\
\end{align*}
\]
\text{from corr-q ie1 PC-monotone have posX: } 0 \leq ?X
from IClock corr-q have \(?T1 = cfn q (\vartheta q (i+1)) + ?X\)
  by(simp add: Adj-def)

from this posX trans-inv show ?thesis by simp
qed

**lemma** beta-rho:
**assumes** ie: \(te q (i+1) \leq te p (i+1)\)
and corr-p: correct p \((te p (i+1))\)
and corr-q: correct q \((te p (i+1))\)
and corr-l: correct l \((te p (i+1))\)
shows \(\left| (PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1)) \right| \leq \beta \cdot \rho\)

**proof**
  let \(?X = (PC l (te p (i+1)) − PC l (te q (i+1)))\)
  let \(?D = te p (i+1) − te q (i+1)\)

  from ie have posX trans-inv show ?thesis by simp
  qed

**lemma** beta-rho:
**assumes** ie: \(te q (i+1) \leq te p (i+1)\)
and corr-p: correct p \((te p (i+1))\)
and corr-q: correct q \((te p (i+1))\)
and corr-l: correct l \((te p (i+1))\)
shows \(\left| (PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1)) \right| \leq \beta \cdot \rho\)

**proof**
  let \(?X = (PC l (te p (i+1)) − PC l (te q (i+1)))\)
  let \(?D = te p (i+1) − te q (i+1)\)

  from ie have posX trans-inv show ?thesis by simp
  qed

**lemma** beta-rho:
**assumes** ie: \(te q (i+1) \leq te p (i+1)\)
and corr-p: correct p \((te p (i+1))\)
and corr-q: correct q \((te p (i+1))\)
and corr-l: correct l \((te p (i+1))\)
shows \(\left| (PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1)) \right| \leq \beta \cdot \rho\)

**proof**
  let \(?X = (PC l (te p (i+1)) − PC l (te q (i+1)))\)
  let \(?D = te p (i+1) − te q (i+1)\)

  from ie have posX trans-inv show ?thesis by simp
  qed

**lemma** beta-rho:
**assumes** ie: \(te q (i+1) \leq te p (i+1)\)
and corr-p: correct p \((te p (i+1))\)
and corr-q: correct q \((te p (i+1))\)
and corr-l: correct l \((te p (i+1))\)
shows \(\left| (PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1)) \right| \leq \beta \cdot \rho\)

**proof**
  let \(?X = (PC l (te p (i+1)) − PC l (te q (i+1)))\)
  let \(?D = te p (i+1) − te q (i+1)\)

  from ie have posX trans-inv show ?thesis by simp
  qed

**lemma** beta-rho:
**assumes** ie: \(te q (i+1) \leq te p (i+1)\)
and corr-p: correct p \((te p (i+1))\)
and corr-q: correct q \((te p (i+1))\)
and corr-l: correct l \((te p (i+1))\)
shows \(\left| (PC l (te p (i+1)) − PC l (te q (i+1))) − (te p (i+1) − te q (i+1)) \right| \leq \beta \cdot \rho\)

**proof**
  let \(?X = (PC l (te p (i+1)) − PC l (te q (i+1)))\)
  let \(?D = te p (i+1) − te q (i+1)\)

  from ie have posX trans-inv show ?thesis by simp
  qed

This lemma (and the next one pe-cond2) proves an assumption used in the precision enhancement.
and corr-l: correct l (te p (i+1))

shows \(|\varrho \ q \ (i+1) \ l + (PC \ q \ (te \ p \ (i+1)) - PC \ q \ (te \ q \ (i+1))) - \varrho \ p \ (i+1) \ l| \leq 2* \ \varrho * \ \beta + 2* \ \Lambda

(is ?M \leq ?N)

proof –

let ?Xl = (PC l (te p (i+1)) - PC l (te q (i+1)))
let ?Xq = (PC q (te p (i+1)) - PC q (te q (i+1)))
let ?D = te p (i+1) - te q (i+1)
let ?T = \varrho \ p \ (i+1) \ l - \varrho \ q \ (i+1) \ l
let ?RE1 = \varrho \ p \ (i+1) \ l - IC l i (te p (i+1))
let ?RE2 = \varrho \ q \ (i+1) \ l - IC l i (te q (i+1))
let ?ICT = IC l i (te p (i+1)) - IC l i (te q (i+1))

have ?M = |(?Xq - ?D) - (?T - ?D)|
by(simp add: abs-if)

by(simp add: abs-if)

from ie corr-q correct-closed have corr-q-tq: correct q (te q (i+1))
by(blast)

from ie corr-l correct-closed have corr-l-tq: correct l (te q (i+1))
by blast

from corr-p corr-q corr-l ie beta-rho
have XI D: |?XI - ?D| \leq \beta * \varrho
by simp

from corr-p corr-q ie beta-rho
have Xq D: |?Xq - ?D| \leq \beta * \varrho
by simp

have TD: |?T - ?D| \leq 2* \ \Lambda + \beta * \varrho

proof –

by (simp add: abs-if)

by(simp add: abs-if)

have Eq3: |?T - ?ICT| \leq |?RE1| + |?RE2|
by(simp add: abs-if)

from readerror corr-p corr-l
have Eq4: |?RE1| \leq \Lambda
by simp

from corr-l-tq corr-q-tq this readerror
have Eq5: |?RE2| \leq \Lambda
by simp

from Eq3 Eq4 Eq5 have Eq6: |?T - ?ICT| \leq 2* \ \Lambda
by simp

have Eq7: ?ICT - ?D = ?XI - ?D
proof
  from \texttt{corr-p rte} have \( t e \ p \ i \leq t e \ p \ (i+1) \)
    by (simp)
  from this \texttt{corr-l correct-closed} have \texttt{corr-l-tpi}: \( t e \ p \ (t e \ p \ i) \)
    by blast
  from \texttt{corr-q-tg \ rte} have \( t e \ q \ i \leq t e \ q \ (i+1) \)
    by simp
  from this \texttt{corr-l-tq \ correct-closed} have \texttt{corr-l-tqi}: \( t e \ q \ (t e \ q \ i) \)
    by blast

  from \texttt{IClock \ corr-l}
  have \( F1: I C \ l \ i \ (t e \ p \ (i+1)) = P C \ l \ (t e \ p \ (i+1)) + A d j \ l \ i \)
    by (simp)
  from \texttt{IClock \ corr-l-tq}
  have \( F2: I C \ l \ i \ (t e \ q \ (i+1)) = P C \ l \ (t e \ q \ (i+1)) + A d j \ l \ i \)
    by simp
  from \( F1 \ F2 \) show \( \texttt{thesis} \) by (simp)
qed

from \texttt{this XlD} have \( \l E q 8: |\l E q C t - \l E q D| \leq \beta \ast \varrho \)
  by arith
from \( E q 1 \ E q 2 \ E q 6 \ E q 8 \) show \( \texttt{thesis} \)
  by (simp)
qed

from \texttt{Split XqD TD} have \( \l E q 1: \l E M \leq 2 \ast \beta \ast \varrho + 2 \ast \Lambda \)
  by (simp)
have \( \l E q 2: 2 \ast \varrho \ast \beta + 2 \ast \Lambda = 2 \ast \beta \ast \varrho + 2 \ast \Lambda \)
  by simp
from \( \l E q 1 \) show \( \texttt{thesis} \) by (simp only: \( \l E q 2 \))
qed

lemma \texttt{pe-cond2}:
assumes \( i c: t e \ m \ i \leq t e \ l \ i \)
and \texttt{corr-k}: \( \texttt{correct k} \ (t e \ k \ (i+1)) \)
and \texttt{corr-l-tk}: \( \texttt{correct l} \ (t e \ k \ (i+1)) \)
and \texttt{corr-m-tk}: \( \texttt{correct m} \ (t e \ k \ (i+1)) \)
and \texttt{ind-hyp}: \( |I C \ l \ i \ (t e \ l \ i) - I C \ m \ i \ (t e \ l \ i)| \leq \delta \)
shows \( \vartheta \ k \ (i+1) \ l - \vartheta \ k \ (i+1) \ m \leq 2 \ast \Lambda + \delta \ S + 2 \ast \varrho \ast (r m a x + \beta) \)
proof–
  let \( ?X = \vartheta \ k \ (i+1) \ l - \vartheta \ k \ (i+1) \ m \)
  let \( ?N = 2 \ast \Lambda + \delta \ S + 2 \ast \varrho \ast (r m a x + \beta) \)
  let \( ?D1 = \vartheta \ k \ (i+1) \ l - I C \ l \ i \ (t e \ k \ (i+1)) \)
  let \( ?D2 = \vartheta \ k \ (i+1) \ m - I C \ m \ i \ (t e \ k \ (i+1)) \)
  let \( ?I C S = I C \ l \ i \ (t e \ k \ (i+1)) - I C \ m \ i \ (t e \ k \ (i+1)) \)
  let \( ?tlm = t e \ l \ i \)
  let \( ?I C = I C \ l \ i \ ?tlm - I C \ m \ i \ ?tlm \)
  have \( \l E q 1: |?X| = |(?D1 - ?D2) + ?I C S| \) (is \( ?E1 = ?E2 \))
    by (simp add: abs-if)
  have \( \l E q 2: ?E2 \leq |?D1 - ?D2| + |?I C S| \) by (simp add: abs-if)
from corr-l-tk corr-k beta-bound1 have ie-lk: te l i ≤ te k (i+1)
  by (simp add: le-diff-eq)

from this corr-l-tk correct-closed have corr-l: correct l (te l i)
  by blast

from ie-lk corr-l-tk corr-m-tk IC-bd
have Eq3: |?ICS| ≤ |?IC| + 2*ϱ*(te k (i+1) - ?tlm)
  by simp
from this ind-hyp have Eq4: |?ICS| ≤ δS + 2*ϱ*(te k (i+1) - ?tlm)
  by simp

from corr-l corr-k beta-bound2 have te k (i+1) - ?tlm ≤ rmax + β
  by simp
from this constants-ax have 2*ϱ*((te k (i+1) - ?tlm) ≤ 2*ϱ*(rmax + β)
  by (simp add: real-mult-le-cancel-iff2)
from this Eq4 have Eq4a: |?ICS| ≤ δS + 2*ϱ*(rmax + β)
  by (simp)

from corr-k corr-l-tk readerror
have Eq5: |?D1| ≤ Λ by simp
from corr-k corr-m-tk readerror
have Eq6: |?D2| ≤ Λ by simp
have |?D1 - ?D2| ≤ |?D1| + |?D2| by (simp add: abs-if)
from this Eq5 Eq6 have Eq7: |?D1 - ?D2| ≤ 2*Λ
  by (simp)

from Eq1 Eq2 Eq4a Eq7 split show ?thesis by (simp)
qed

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))| ≤ δS
shows |φ p (i+1) l - φ p (i+1) m| ≤ 2*Λ + δS + 2*ϱ*(rmax + β)
proof–
from corr-p corr-l beta-bound1 have tli-le-tp: te l i ≤ te p (i+1)
  by (simp add: le-diff-eq)
from corr-p corr-m beta-bound1 have tmi-le-tp: te m i ≤ te p (i+1)
  by (simp add: le-diff-eq)

let ?tml = max (te l i) (te m i)
from tli-le-tp tmi-le-tp have tml-le-tp: ?tml ≤ te p (i+1)
  by simp

from tml-le-tp corr-l correct-closed have corr-l-tml: correct l ?tml
  by blast
from tml-le-tp corr-m correct-closed have corr-m-tml: correct m ?tml
by blast

let ?Y = 2*λ + δS + 2*ϱ*(rmax + β)
show |ϑ p (i+1) l - ϑ p (i+1) m| ≤ ?Y

proof cases
  assume A: te m i < te l i
  from this IC-bound
  have |IC l i (te l i) - IC m i (te l i)| ≤ δS
    by (simp add: max-def)
  from this A corr-p corr-l corr-m pe-cond2
  show ?thesis by (simp)
  next
  assume ¬ (te m i < te l i)
  hence Eq1: te l i ≤ te m i by simp
  from this IC-bound
  have Eq2: |IC l i (te m i) - IC m i (te m i)| ≤ δS
    by (simp add: max-def)
  hence |IC m i (te m i) - IC l i (te m i)| ≤ δS
    by (simp add: abs-minus-commute)
  from this Eq1 corr-p corr-l corr-m pe-cond2
  have |ϑ p (i+1) m - ϑ p (i+1) l| ≤ ?Y
    by (simp)
  thus ?thesis by (simp add: abs-minus-commute)
qed

lemma four-one-ind-half:
  assumes ie1: β ≤ rmin
  and ie2: µ ≤ δS
  and ie3: γ1 δS ≤ δS
  and ind-hyp: okmaxsync i δS
  and ie4: 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 p (i+1) (te p (i+1)) - IC q (i+1) (te p (i+1))| ≤ δS

proof
  let ?tpq = te p (i+1)
  from ie4 corr-q correct-closed have corr-q-tq: correct q (te q (i+1))
    by blast
  have Eq-IC-cfn: |IC p (i+1) ?tpq - IC q (i+1) ?tpq| =
    |cfn q ?f - cfn p ?g| |
  proof
    from corr-p ICp-Suc have Eq1: IC p (i+1) ?tpq = cfn p ?g by simp
  from ie4 corr-p corr-q IC-trans-inv
  have Eq2: IC q (i+1) ?tpq = cfn q ?f by simp
from Eq1 Eq2 show ?thesis by (simp add: abs-if)
qed

let ?ppred = λ l. correct l (te p (i+1))

let ?X = 2*g*β + 2*Λ

have ∀ l. ?ppred l → |?f l - ?g l| ≤ ?X

proof –
{
  fix l
  assume ?ppred l
  from ie4 corr-p corr-q this pe-cond1
  have |?f l - ?g l| ≤ (2*g*β + 2*Λ)
    by (auto)
}

thus ?thesis by blast

qed

hence cond1: okRead2 ?f ?g ?X ?ppred
by (simp add: okRead2-def)

let ?Y = 2*Λ + δS + 2*g*(rmax + β)

have ∀ l m. ?ppred l ∧ ?ppred m → |?f l - ?f m| ≤ ?Y

proof –
{
  fix l m
  assume corr-l: ?ppred l
  assume corr-m: ?ppred m

  from corr-p corr-l beta-bound1 have tli-le-tp: te l i ≤ te p (i+1)
    by (simp add: le-diff-eq)
  from corr-p corr-m beta-bound1 have tmi-le-tp: te m i ≤ te p (i+1)
    by (simp add: le-diff-eq)

  let ?tlm = max (te l i) (te m i)

  from tli-le-tp tmi-le-tp have tlm-le-tp: ?tlm ≤ te p (i+1)
    by simp

  from ie4 corr-l correct-closed have corr-l-tq: correct l (te q (i+1))
    by blast
  from ie4 corr-m correct-closed have corr-m-tq: correct m (te q (i+1))
    by blast
  from tlm-le-tp corr-l correct-closed have corr-l-tlm: correct l ?tlm
    by blast
  from tlm-le-tp corr-m correct-closed have corr-m-tlm: correct m ?tlm
    by blast

  from ind-hyp corr-l-tlm corr-m-tlm
  have EqAbs1: |IC l i ?tlm - IC m i ?tlm| ≤ δS
    by (auto simp add: okmaxsync-def)
have EqAbs3: |?f l − ?f m| = |? q (i+1) l − ? q (i+1) m|
    by (simp add: abs-if)

from EqAbs1 corr-q-tq corr-l-tq corr-m-tq theta-bound
have |? q (i+1) l − ? q (i+1) m| ≤ ?Y
    by simp
from this EqAbs3 have |?f l − ?f m| ≤ ?Y
    by simp

} thus ?thesis by simp

qed

hence cond2a: okRead1 ?f ?Y ?ppred by (simp add: okRead1-def)

have ∀ l m. ?ppred l ∧ ?ppred m → |?g l − ?g m| ≤ ?Y
proof −
{ fix l m
  assume corr-l: ?ppred l
  assume corr-m: ?ppred m

  from corr-p corr-l beta-bound1 have tli-le-tp: te l i ≤ te p (i+1)
    by (simp add: le-diff-eq)
  from corr-p corr-m beta-bound1 have tmi-le-tp: te m i ≤ te p (i+1)
    by (simp add: le-diff-eq)

  let ?tlm = max (te l i) (te m i)
  from tli-le-tp tmi-le-tp have tlm-le-tp: ?tlm ≤ te p (i+1)
    by simp

  from tlm-le-tp corr-l correct-closed have corr-l-tlm: correct l ?tlm
    by blast
  from tlm-le-tp corr-m correct-closed have corr-m-tlm: correct m ?tlm
    by blast

  from ind-hyp corr-l-tlm corr-m-tlm
  have EqAbs1: |IC l i ?tlm − IC m i ?tlm| ≤ δS
    by(auto simp add: okmaxsync-def)

  from EqAbs1 corr-p corr-l corr-m theta-bound
  have |?g l − ?g m| ≤ ?Y by simp

  } thus ?thesis by simp

qed

hence cond2b: okRead1 ?g ?Y ?ppred by (simp add: okRead1-def)

from correct-count have np − maxfaults ≤ count ?ppred np
    by simp
from this corr-p corr-q cond1 cond2a cond2b prec-enh
have |cfn q ?f − cfn p ?g| ≤ π ?X ?Y
    by blast

from ie3 this have |cfn q ?f − cfn p ?g| ≤ δS
    by (simp add: γ1-def)
from this Eq-IC-cfn show \( \text{thesis} \) by (simp)

\[
\text{qed}
\]

Theorem 4.1 in Shankar’s paper.

**theorem** four-one:
- assumes \( ie1: \beta \leq r_{\text{min}} \)
- and \( ie2: \mu \leq \delta S \)
- and \( ie3: \gamma 1 \delta S \leq \delta S \)
- shows \( \text{okmaxsync} \ i \ \delta S \)

**proof** (induct \( i \))
- show \( \text{okmaxsync} \ 0 \ \delta S \)

\[
\begin{align*}
\text{proof} & - \\
& \{ \\
& \text{fix} \ p \ q \\
& \text{assume} \ corr-p: \ \text{correct} \ p \ (\max (te p 0) (te q 0)) \\
& \text{assume} \ corr-q: \ \text{correct} \ q \ (\max (te p 0) (te q 0)) \\
& \text{from} \ corr-p \ \text{synch0} \ have \ cp0: \ \text{correct} \ p \ 0 \ \text{by simp} \\
& \text{from} \ corr-q \ \text{synch0} \ have \ cq0: \ \text{correct} \ q \ 0 \ \text{by simp} \\
& \text{from} \ \text{synch0} \ cp0 \ cq0 \ \text{IClock} \\
& \text{have} \ \text{IC-} \text{eq-PC} : \\
& \ |IC p 0 \ (\max (te p 0) (te q 0)) - IC q 0 \ (\max (te p 0) (te q 0))| \\
& = |PC p 0 - PC q 0| \ (\text{is} \ T1 = \ T2) \\
& \text{by} (\text{simp} \ \text{add}: \ \text{Adj-def}) \\
& \text{from} \ ie2 \ \text{init} \ \text{synch0} \ cp0 \ \text{have} \ \text{range1}: \ 0 \leq PC p 0 \land PC p 0 \leq \delta S \ \text{by auto} \\
& \text{from} \ ie2 \ \text{init} \ \text{synch0} \ cq0 \ \text{have} \ \text{range2}: \ 0 \leq PC q 0 \land PC q 0 \leq \delta S \ \text{by auto} \\
& \text{have} \ T2 \leq \delta S \\
& \text{proof} \ \text{cases} \\
& \text{assume} \ \text{A}: PC p 0 < PC q 0 \\
& \text{from} \ A \ \text{range1 range2} \ show \ \text{thesis} \\
& \text{by} (\text{auto} \ \text{simp} \ \text{add}: \ \text{abs-if}) \\
& \text{next} \\
& \text{assume} \ \text{notA}: = (PC p 0 < PC q 0) \\
& \text{from} \ \text{notA range1 range2} \ show \ \text{thesis} \\
& \text{by} (\text{auto} \ \text{simp} \ \text{add}: \ \text{abs-if}) \\
& \text{qed} \\
& \text{from} \ this \ \text{IC-} \text{eq-PC} \ \text{have} \ \text{T1} \leq \delta S \ \text{by simp} \\
& \} \\
& \text{thus} \ \text{thesis} \ \text{by} (\text{simp} \ \text{add}: \ \text{okmaxsync-def}) \\
& \text{qed} \\
\text{next} \\
\text{fix} \ i \ \text{assume} \ \text{ind-hyp}: \ \text{okmaxsync} \ i \ \delta S \\
\text{show} \ \text{okmaxsync} \ (Suc \ i) \ \delta S \\
\text{proof} - \\
& \{ \\
& \text{fix} \ p \ q \\
& \text{assume} \ corr-p: \ \text{correct} \ p \ (\max (te p (i + 1)) (te q (i + 1))) \\
& \text{assume} \ corr-q: \ \text{correct} \ q \ (\max (te p (i + 1)) (te q (i + 1))) \\
\}
\]

19
let \( ?tp = \text{te } p(i+1) \)
let \( ?tq = \text{te } q(i+1) \)
let \( ?tpq = \max(?tp ?tq) \)

have \(|IC p(i+1) ?tpq - IC q(i+1) ?tpq| \leq \delta S\) (is \(?E1 \leq \delta S\))
proof cases
assume A: \(?tq < ?tp\)
from A corr-p have cp1: correct p \((\text{te } p(i+1))\)
  by (simp add: max-def)
from A corr-q have cq1: correct q \((\text{te } p(i+1))\)
  by (simp add: max-def)
from A
have Eq1: \(?E1 = |IC p(i+1) (\text{te } p(i+1)) - IC q(i+1) (\text{te } p(i+1))|\) (is \(?E1 = ?E2\))
  by (simp add: max-def)
from A cp1 cq1 corr-p corr-q ind-hyp ie1 ie2 ie3
  four-one-ind-half
have \(?E2 \leq \delta S\) by (simp)
from this Eq1 show \(?thesis\) by simp
next
assume notA: \(\neg(\text{?tq < ?tp})\)
from this corr-p have cp2: correct p \((\text{te } q(i+1))\)
  by (simp add: max-def)
from notA corr-q have cq2: correct q \((\text{te } q(i+1))\)
  by (simp add: max-def)
from notA
have Eq2: \(?E1 = |IC q(i+1) (\text{te } q(i+1)) - IC p(i+1) (\text{te } q(i+1))|\) (is \(?E1 = ?E3\))
  by (simp add: max-def abs-minus-commute)
from notA have \(?tp \leq ?tq\) by simp
from this cp2 cq2 ind-hyp ie1 ie2 ie3 four-one-ind-half
have \(?E3 \leq \delta S\) by simp
from this Eq2 show \(?thesis\) by (simp)
qed

thus \(?thesis\) by (simp add: okmaxsync-def)
qed

lemma VC-cfn:
  assumes corr-p: correct p \((\text{te } p(i+1))\)
  and ie: \(\text{te } p(i+1) < \text{te } p(i+2)\)
shows VC p \((\text{te } p(i+1)) = \text{cfn} p (\varnothing p (i+1))\)
proof
from ie corr-p VClock have VC p \((\text{te } p(i+1)) = IC p(i+1) (\text{te } p(i+1))\)
  by simp
moreover
from corr-p IClock
have IC p \((i+1) (\text{te } p(i+1)) = PC p(\text{te } p(i+1)) + Adj p(i+1)\)
  by blast
moreover
have PC p \((\text{te } p(i+1)) + Adj p(i+1) = \text{cfn} p (\varnothing p(i+1))\)
by (simp add: Adj-def)
ultimately show ?thesis by simp
qed

Lemma for the inductive case in Theorem 4.2

lemma four-two-ind:
  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: \( t_{e q (i+1)} \leq t_{e p (i+1)} \)
  and ind-hyp: okClocks p q i
  and t-bound1: \( 0 \leq t \)
  and t-bound2: \( t < \max (t_{e p (i+1)}) (t_{e q (i+1)}) \)
  and t-bound3: \( \max (t_{e p i}) (t_{e q i}) \leq t \)
  and tpq-bound: \( \max (t_{e p i}) (t_{e q i}) < \max (t_{e p (i+1)}) (t_{e q (i+1)}) \)
  and corr-p: correct p t
  and corr-q: correct q t
shows \( |VC p t - VC q t| \leq \delta \)
proof cases
  assume A: \( t < t_{e q (i+1)} \)
  let \( \?tpq = \max (t_{e p i}) (t_{e q i}) \)
  have Eq1: \( t_{e p i} \leq t \wedge t_{e q i} \leq t \)
  proof cases
    assume te p i \( \leq \) te q i
    from this t-bound3 show ?thesis by (simp add: max-def)
  next
    assume \( \neg (t_{e p i} \leq t_{e q i}) \)
    from this t-bound3 show ?thesis by (simp add: max-def)
  qed

from ie6 have tp-max: \( \max (t_{e p (i+1)}) (t_{e q (i+1)}) = t_{e p (i+1)} \)
  by (simp add: max-def)
from this t-bound2 have Eq2: \( t < t_{e p (i+1)} \) by simp

from VClock Eq1 Eq2 corr-p have Eq3: \( VC p t = IC p i t \) by simp
from VClock Eq1 A corr-q have Eq4: \( VC q t = IC q i t \) by simp
from Eq3 Eq4 have Eq5: \( |VC p t - VC q t| = |IC p i t - IC q i t| \)
  by simp

from t-bound3 corr-p corr-q correct-closed
have corr-tpq: correct p ?tpq \wedge correct q ?tpq
  by (blast)

from t-bound3 IC-bd corr-p corr-q
have Eq6: \( |IC p i t - IC q i t| \leq |IC p i ?tpq - IC q i ?tpq| + 2 \cdot \rho \cdot (t - ?tpq) \) (is \( ?E1 \leq ?E2 \))
  by (blast)
from \texttt{ie1 ie2 ie3 four-one} have \texttt{okmaxsync i} \(\delta S\) \texttt{by simp}

from \texttt{this corr-tpq} have \(|IC\ p\ i\ \bot_{\text{tpq}} - IC\ q\ i\ \bot_{\text{tpq}}| \leq \delta S\)
by (simp add: \texttt{okmaxsync-def})

from \texttt{Eq6 this have} \(\texttt{Eq7:}\ \bot_{\text{tpq}} \leq \delta S + 2^*\varrho^*(t - \bot_{\text{tpq}})\) \texttt{by simp}

from \texttt{corr-p Eq2 rts0 have} \(t - te\ p\ i \leq rmax\) \texttt{by simp}
from \texttt{this have} \(t - \bot_{\text{tpq}} \leq rmax\) \texttt{by (simp add: max-def)}
from \texttt{this constants-ax have} \(2^*\varrho^*(t - \bot_{\text{tpq}}) \leq 2^*\varrho^*rmax\)
by (simp add: real-mult-le-cancel-iff1)

from \texttt{this Eq7 have} \(\texttt{Eq8:}\ \bot_{\text{tpq}} \leq \delta S + 2^*\varrho^*rmax\) \texttt{by simp}
from \texttt{this Eq5 ie4 show} \(\texttt{thesis by simp add: \gamma2-def})

next
assume \(\neg(t < te\ q\ (i+1))\)

hence \(B: te\ q\ (i+1) \leq t\) \texttt{by simp}

from \texttt{ie6 t-bound2 have} \(tp-max; max\ (te\ p\ (i+1)) (te\ q\ (i+1)) = te\ p\ (i+1)\)
by (simp add: max-def)

have \(te\ p\ i \leq max\ (te\ p\ i) (te\ q\ i)\)
by (simp add: max-def)

from \texttt{this t-bound3 have} \(tp-bound1: te\ p\ i \leq t\) \texttt{by simp}

from \texttt{tp-max t-bound2 have} \(tp-bound2: t < te\ p\ (i+1)\) \texttt{by simp}

have \(tq-bound1: t < te\ q\ (i+2)\)
\texttt{proof (rule ccontr)}

assume \(\neg(t < te\ q\ (i+2))\)

hence \(C: te\ q\ (i+2) \leq t\) \texttt{by simp}

from \texttt{C corr-q correct-closed have} \(corr-q-t2: correct\ q\ (te\ q\ (i+2))\) \texttt{by blast}

have \(te\ q\ (i+1) + \beta \leq t\)
\texttt{proof}

from \texttt{corr-q-t2 rts1d have} \(rmin \leq te\ q\ (i+2) - te\ q\ (i+1)\)
by (simp)

from \texttt{this ie1 have} \(\beta \leq te\ q\ (i+2) - te\ q\ (i+1)\)
by (simp)

hence \(te\ q\ (i+1) + \beta \leq te\ q\ (i+2)\) \texttt{by simp}

from \texttt{this C show} \(\texttt{thesis by simp}\)
qed

from \texttt{this corr-p corr-q rts2a have} \(te\ p\ (i+1) \leq t\)
by blast

hence \(\neg(t < te\ p\ (i+1))\) \texttt{by simp}

from \texttt{this tp-bound2 show} \(\texttt{False by simp}\)
qed
from \( \text{tq-bound1} \) \( B \) have \( \text{tq-bound2} \): \( \text{te} q \ (i+1) < \text{te} q \ (i+2) \) by simp
from \( B \) \( \text{tp-bound2} \) have \( \text{tq-bound3} \): \( \text{te} q \ (i+1) < \text{te} p \ (i+1) \) by simp
from \( B \) \( \text{corr-p correct-closed} \)
have \( \text{corr-p-tq1} \): \( \text{correct} \ p \ (\text{te} q \ (i+1)) \) by blast
from \( B \) \( \text{corr-p} \) \( \text{corr-q} \) \( \text{corr-p-tq1 correct-closed} \)
have \( \text{corr-q-tq1} \): \( \text{correct} \ q \ (\text{te} q \ (i+1)) \) by blast
from \( \text{corr-p-tq1 corr-q-tq1 beta-bound1} \)
have \( \text{tq-bound4} \): \( \text{te} p \ i \leq \text{te} q \ (i+1) \) by (simp add: le-diff-eq)
from \( \text{tq-bound1 VClock B corr-q} \)
have \( \text{Eq1} \): \( \text{VC} \ q \ t = \text{IC} \ q \ (i+1) \ t \) by simp
from \( \text{VClock tp-bound1 tp-bound2 corr-p} \)
have \( \text{Eq2} \): \( \text{VC} \ p \ t = \text{IC} \ p \ i \ t \) by simp
from \( \text{Eq1 Eq2} \)
have \( \text{Eq3} \): \( |\text{VC} \ p \ t - \text{VC} \ q \ t| = |\text{IC} \ p \ i \ t - \text{IC} \ q \ (i+1) \ t| \)
by simp
from \( \text{this Eq3} \)
have \( \text{VC-split} \): \( |\text{VC} \ p \ t - \text{VC} \ q \ t| \leq |\text{IC} \ p \ i \ (\text{te} q \ (i+1)) - \text{IC} \ q \ (i+1) \ (\text{te} q \ (i+1))| + 2*\rho*(t - \text{te} q \ (i+1)) \)
by simp
from \( \text{tq-bound2 VClock corr-q-tq1} \)
have \( \text{Eq4} \): \( \text{VC} \ q \ (\text{te} q \ (i+1)) = \text{IC} \ q \ (i+1) \ (\text{te} q \ (i+1)) \) by simp
from \( \text{this tq-bound2 VClock corr-q-tq1} \)
have \( \text{Eq5} \): \( \text{IC} \ q \ (i+1) \ (\text{te} q \ (i+1)) = \text{cfn} \ q \ (\vartheta \ q \ (i+1)) \) by simp
hence \( \text{ic-ccf-cfn}: \text{IC} \ p \ i \ (\text{te} q \ (i+1)) - \text{IC} \ q \ (i+1) \ (\text{te} q \ (i+1)) = \text{IC} \ p \ i \ (\text{te} q \ (i+1)) - \text{cfn} \ q \ (\vartheta \ q \ (i+1)) \)
(is \( ?E1 = ?E2 \))
by simp
let \( ?f = \vartheta \ q \ (i+1) \)
let \( ?ppred = \lambda \ l. \text{correct} \ l \ (\text{te} q \ (i+1)) \)
let \( ?X = 2*\Lambda + \delta S + 2*\rho*(r_{\text{max}} + \beta) \)
have \( \forall \ l \ m. \ ?ppred \ l \land \ ?ppred \ m \rightarrow |\vartheta \ q \ (i+1) \ l - \vartheta \ q \ (i+1) \ m| \leq ?X \)
proof -
\{
fix \ l :: \text{process}
fix \ m :: \text{process}
assume \( \text{corr-l}: \ ?ppred \ l \)
assume \( corr-m: \ ?ppred \ m \)

let \( \?tlm = \max (te \ l \ i) (te \ m \ i) \)

have \( tlm\text{-bound}: \ ?tlm \leq te \ q \ (i+1) \)

proof –
  from \( corr-l \ corr-q-tq1 \ beta\text{-bound1} \) have \( te \ l \ i \leq te \ q \ (i+1) \)
  by \( \text{(simp add: le-diff-eq)} \)
moreover
  from \( corr-m \ corr-q-tq1 \ beta\text{-bound1} \) have \( te \ m \ i \leq te \ q \ (i+1) \)
  by \( \text{(simp add: le-diff-eq)} \)
ultimately show \( ?\thesis \) by simp

qed

from \( tlm\text{-bound} \ corr-l \ corr-m \ correct\text{-closed} \)
have \( corr\text{-tlm}: \ correct \ l \ ?tlm \wedge correct \ m \ ?tlm \)
  by \( \text{blast} \)

have \( |IC \ l \ i \ ?tlm - IC \ m \ i \ ?tlm| \leq \delta S \)
proof –
  from \( ie1 \ ie2 \ ie3 \ four\text{-one} \) have \( okmaxsync \ i \ \delta S \)
  by \( \text{simp} \)
from \( this \ corr\text{-tlm} \) show \( ?\thesis \) by(\( \text{simp add: okmaxsync\text{-def}} \))

qed

from \( this \ corr-l \ corr-m \ corr-q-tq1 \ theta\text{-bound} \)
have \( |\vartheta \ q \ (i+1) \ l - \vartheta \ q \ (i+1) \ m| \leq \ ?X \) by simp
\}

thus \( ?\thesis \) by \( \text{blast} \)

qed

hence \( readOK: \ okRead1 \ (\vartheta \ q \ (i+1)) \ ?X \ ?ppred \)
  by(\( \text{simp add: okRead1\text{-def}} \))

let \( \?E3 = cfn \ q \ (\vartheta \ q \ (i+1)) - \vartheta \ q \ (i+1) \ p \)
let \( \?E4 = \vartheta \ q \ (i+1) \ p - IC \ p \ i \ (te \ q \ (i+1)) \)

have \( |?E2| = |?E3 + ?E4| \) by(\( \text{simp add: abs-if} \))

hence \( Eq8: \ |?E2| \leq |?E3| + |?E4| \) by(\( \text{simp add: abs-if} \))

from \( correct\text{-count} \) have \( ppredOK: np - maxfaults \leq count \ ?ppred \ np \)
  by \( \text{simp} \)
from \( readOK \ ppredOK \ corr\text{-p-tq1} \ corr\text{-q-tq1} \ acc\text{-prsv} \)
have \( |?E3| \leq \alpha \ ?X \)
  by \( \text{blast} \)
from \( this \ Eq8 \) have \( Eq9: \ |?E2| \leq \alpha \ ?X + |?E4| \) by\( \text{simp} \)

from \( corr\text{-p-tq1} \ corr\text{-q-tq1} \ readerror \)
have \( |?E4| \leq \Lambda \) by \( \text{simp} \)

from \( this \ Eq9 \) have \( Eq10: \ |?E2| \leq \alpha \ ?X + \Lambda \) by\( \text{simp} \)

from \( this \ VC\text{-split} \ IC\text{-eq-cfn} \)
have \( \text{almost-right:} \)
  \( |VC \ p \ t - VC \ q \ t| \leq \)
Theorem 4.2 in Shankar’s paper.

\begin{proof}
\begin{itemize}
\item \textbf{assumes ie1: } \beta \leq \text{rmin}
\item \textbf{and ie2: } \mu \leq \delta S
\item \textbf{and ie3: } \gamma 1 \delta S \leq \delta S
\item \textbf{and ie4: } \gamma 2 \delta S \leq \delta
\item \textbf{and ie5: } \gamma 3 \delta S \leq \delta
\end{itemize}
\textbf{shows okClocks p q i}
\textbf{proof (induct i)}
\begin{itemize}
\item \textbf{show okClocks p q 0}
\begin{itemize}
\item \textbf{fix t :: time}
\item \textbf{assume t-bound1: } 0 \leq t
\item \textbf{assume t-bound2: } t < \text{max (te p 0) (te q 0)}
\item \textbf{assume corr-p: } correct p t
\item \textbf{assume corr-q: } correct q t
\item \textbf{from t-bound2 synch0 have } t < 0
\item \textbf{by (simp add: max-def)}
\item \textbf{from this t-bound1 have } False \textbf{ by simp}
\item \textbf{hence } |VC p t - VC q t| \leq \delta \textbf{ by simp}
\end{itemize}
\item \textbf{thus } ?thesis \textbf{ by (simp add: okClocks-def)}
\end{itemize}
\textbf{qed}
\textbf{next}
\item \textbf{fix i::nat assume ind-hyp: } okClocks p q i
\item \textbf{show okClocks p q (Suc i)}
\end{proof}
proof –
{
  fix t :: time
  assume t-bound1: 0 ≤ t
  assume t-bound2: t < max (te p (i+1)) (te q (i+1))
  assume corr-p: correct p t
  assume corr-q: correct q t

  let ?tpq1 = max (te p i) (te q i)
  let ?tpq2 = max (te p (i+1)) (te q (i+1))

  have |VC p t − VC q t| ≤ δ
  proof cases
    assume tpq-bound: ?tpq1 < ?tpq2
    show ?thesis
      proof cases
        assume t < ?tpq1
        from t-bound1 this corr-p corr-q ind-hyp
        show ?thesis
          by (simp add: okClocks-def)
      next
        assume ¬ (t < ?tpq1)
        hence tpq-le-t: ?tpq1 ≤ t by arith
        show ?thesis
          proof cases
            assume A: te q (i+1) ≤ te p (i+1)
            from this tpq-le-t tpq-bound ie1 ie2 ie3 ie4 ie5
            ind-hyp t-bound1 t-bound2
            corr-p corr-q tpq-bound four-two-ind
            show ?thesis
              by (simp)
          next
        assume ¬ (te q (i+1) ≤ te p (i+1))
        hence B: te p (i+1) ≤ te q (i+1) by simp
        from ind-hyp okClocks-sym have ind-hyp1: okClocks q p i
          by blast
        have maxsym1: max (te p (i+1)) (te q (i+1)) = max (te q (i+1)) (te p (i+1))
          by (simp add: max-def)
        have maxsym2: max (te p i) (te q i) = max (te q i) (te p i)
          by (simp add: max-def)
        from maxsym1 t-bound2
        have t-bound21: t < max (te q (i+1)) (te p (i+1))
          by simp
        from maxsym1 maxsym2 tpq-bound
        have tpq-bound1: max (te q i) (te p i) < max (te q (i+1)) (te p (i+1))
          by simp
        from maxsym2 tpq-le-t
        have tpq-le-t1: max (te q i) (te p i) ≤ t by simp
    }
from B tpq-le-t1 tpq-bound1 ie1 ie2 ie3 ie4 ie5
  ind-hyp1 t-bound1 t-bound21
corr-p corr-q tpq-bound four-two-ind
have |VC q t − VC p t| ≤ δ by (simp)
thus ?thesis by (simp add: abs-minus-commute)
qed

qed
next
  assume ¬ (?tpq1 < ?tpq2)
hence ?tpq2 ≤ ?tpq1 by arith
from t-bound2 this have t < ?tpq1 by arith
from t-bound1 this corr-p corr-q ind-hyp
show ?thesis by (simp add: okClocks-def)
qed

} thus ?thesis by (simp add: okClocks-def)
qed
qed

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

theorem agreement:
  assumes ie1: β ≤ rmin
  and ie2: μ ≤ δS
  and ie3: γ1 δS ≤ δS
  and ie4: γ2 δS ≤ δ
  and ie5: γ3 δS ≤ δ
  and ie6: 0 ≤ t
  and cpq: correct p t ∧ correct q t
shows |VC p t − VC q t| ≤ δ
proof −
  from ie6 cpq event-bound have ∃ i :: nat. t < max (te p i) (te q i)
    by simp
  from this obtain i :: nat where t-bound: t < max (te p i) (te q i) ..
  from t-bound ie1 ie2 ie3 ie4 ie5 four-two have okClocks p q i
    by simp
  from ie6 this t-bound cpq show ?thesis
    by (simp add: okClocks-def)
qed
end

References

