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

Alwen Tiu
August 16, 2018

Abstract

We formalize the generalized Byzantine fault-tolerant clock synchronization protocol of Schneider. This protocol abstracts from particular algorithms or implementations for clock synchronization. This abstraction includes several assumptions on the behaviors of physical clocks and on general properties of concrete algorithms/implementations. Based on these assumptions the correctness of the protocol is proved by Schneider. His proof was later verified by Shankar using the theorem prover EHDM (precursor to PVS). Our formalization in Isabelle/HOL is based on Shankar’s formalization.

Contents

1 Introduction 1
2 Isar proof scripts 2
   2.1 Types and constants definitions 2
   2.2 Clock conditions 3
      2.2.1 Some derived properties of clocks 5
      2.2.2 Bounded-drift for logical clocks (IC) 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.

type-synonym process = nat
type-synonym event = nat
type-synonym time = real
type-synonym Clocktime = real

axiomatization
\[ \delta :: \text{real and} \]
\[ \mu :: \text{real and} \]
\[ \rho :: \text{real and} \]
\[ rmin :: \text{real and} \]
\[ rmax :: \text{real and} \]
\[ \beta :: \text{real and} \]
\[ \Lambda :: \text{real and} \]
\[ np :: \text{process and} \]
\[ maxfaults :: \text{process and} \]

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

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

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

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

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

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

**definition**

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

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

**definition**

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

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

**definition**

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

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

**definition**

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

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

**definition**

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

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

**definition**

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

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

### 2.2 Clock conditions

Some general assumptions

**axiomatization where**

\[ \text{constants-ax: } 0 < \beta \land 0 < \mu \land 0 < r_{\text{min}} \]

\[ \land r_{\text{min}} \leq r_{\text{max}} \land 0 < \varrho \land 0 < n p \land \text{maxfaults} \leq n p \]

**axiomatization where**

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

**axiomatization where**

\[ \text{VClock: } \forall p t i. \text{correct } p t \land te p i \leq t \land t < te p (i + 1) \longrightarrow VC p t = IC p i t \]
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)∗(1 + ϱ) and
rate-2: ∀ p s t. correct p t ∧ s ≤ t ➝ (t − s)∗(1 − ϱ) ≤ PC p t − PC p s

Condition 3: bounded interval
axiomatization where
rts0: ∀ p t i. correct p t ∧ t ≤ te p (i+1) ➝ t − te p i ≤ rmax and
rts1: ∀ p t i. correct p t ∧ te p (i+1) ≤ t ➝ rmin ≤ t − te p i

Condition 4: bounded delay
axiomatization where
rts2a: ∀ p q t i. correct p t ∧ correct q t ∧ te q i + β ≤ t ➝ te p i ≤ t and
rts2b: ∀ p q t i. correct p (te p i) ∧ correct q (te q i) ➝ abs(te p i − te q i) ≤ β

Condition 5: initial synchronization
axiomatization where
synch0: ∀ p. te p 0 = 0

Condition 6: nonoverlap
axiomatization where
nonoverlap: β ≤ rmin

Condition 7: reading errors
axiomatization where
readererr: ∀ p q i. correct p (te p (i+1)) ∧ correct q (te p (i+1)) ➝
abs(ϑ p (i+1) q − IC q i (te p (i+1))) ≤ Λ

Condition 8: bounded faults
axiomatization where
correct-closed: ∀ p s t. s ≤ t ∧ correct p t ➝ correct p s and
correct-count: ∀ t. np − maxfaults ≤ count λ p. correct p t np

Condition 9: Translation invariance
axiomatization where
trans-inv: ∀ p f x. 0 ≤ x ➝ cfn p (λ y. f y + x) = cfn p f + x

Condition 10: precision enhancement
axiomatization where
prec-enh:
∀ ppred p q f g x y.
np − maxfaults ≤ count ppred np ∧
okRead1 f y ppred ∧ okRead1 g y ppred ∧
\[ \text{okRead2} \, f \, g \, x \, \text{ppred} \land \text{ppred} \, p \land \text{ppred} \, q \rightarrow \text{abs}(\text{cfn} \, p \, f - \text{cfn} \, q \, g) \leq \pi \, x \, y \]

Condition 11: accuracy preservation

axiomatization where

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

2.2.1 Some derived properties of clocks

lemma rts0d:
assumes \text{cp}: correct \, p \,(te \, p \,(i+1))
shows \(te \, p \,(i+1) - te \, p \, i \leq \text{rmax}\)
using \text{cp} \, \text{rts0} \, \text{by simp}

lemma rts1d:
assumes \text{cp}: correct \, p \,(te \, p \,(i+1))
shows \(rmin \leq te \, p \,(i+1) - te \, p \, i\)
using \text{cp} \, \text{rts1} \, \text{by simp}

lemma rte:
assumes \text{cp}: correct \, p \,(te \, p \,(i+1))
shows \(te \, p \, i \leq te \, p \,(i+1)\)
proof–
from \text{cp} \, \text{rts1d} \, \text{have} \, rmin \leq te \, p \,(i+1) - te \, p \, i \text{ by simp}
from this \text{constants-ax} \, \text{show} \, ?\text{thesis} \, \text{by arith}\nqed

lemma beta-bound1:
assumes \text{corr-p}: correct \, p \,(te \, p \,(i+1))
and \text{corr-q}: correct \, q \,(te \, p \,(i+1))
shows \(0 \leq te \, p \,(i+1) - te \, q \, i\)
proof–
from \text{corr-p} \, \text{rte} \, \text{have} \, te \, p \, i \leq te \, p \,(i+1) \text{ by simp}
from this \text{corr-p} \, \text{correct-closed} \, \text{have} \, \text{corr-pi}: \, \text{correct} \, p \,(te \, p \, i) \text{ by blast}
from \text{corr-p} \, \text{rts1d} \, \text{nonoverlap} \, \text{have} \, rmin \leq te \, p \,(i+1) - te \, p \, i \text{ by simp}
from this \text{nonoverlap} \, \text{have} \, \beta \leq te \, p \,(i+1) - te \, p \, i \text{ by simp}
hence \, te \, p \, i + \beta \leq te \, p \,(i+1) \text{ by simp}
from this \text{corr-p} \, \text{corr-q} \, \text{rts2a}
\, \text{have} \, te \, q \, i \leq te \, p \,(i+1) \text{ by blast}
thus \, ?\text{thesis} \, \text{by simp}\nqed

lemma beta-bound2:
assumes \text{corr-p}: correct \, p \,(te \, p \,(i+1))
and \text{corr-q}: correct \, q \,(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 \( ?\text{thesis} \)
  by (simp add: abs-if)
next
assume \( \neg (te \ q \ i \leq te \ p \ i) \)
from this Eq1 show \( ?\text{thesis} \)
  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 \( ?\text{thesis} \) by simp

qed

2.2.2 Bounded-drift for logical clocks (IC)

**lemma 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 \( \text{corr-p}: \text{correct} \ p \ t \)
and \( \text{corr-q}: \text{correct} \ q \ t \)

shows \( |C \ p \ t - D \ q \ t| \leq |C \ p \ s - D \ q \ s| + 2*g*(t - s) \)

**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 \( |?Bp - ?Bq| \leq 2*g*(t - s) \)

**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: \( ?E2 \leq ?I*(1+\rho) - ?I*(1 - \rho) \)
by(simp)

have Eq4: \( ?I*(1+\rho) - ?I*(1 - \rho) = 2*\rho*?I \)
by(simp add: algebra-simps)

from Eq1 Eq2 Eq3 Eq4 show \( \text{thesis by simp} \)
qed

moreover have \( |?Dt| \leq |?Bp - ?Bq| + |?Ds| \)
by(simp add: abs-if)
ultimately show \( \text{thesis by simp} \)
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 \( |C p t - D q t| \leq |C p s - D q s| + 2*\rho*(t - s) \)
proof–
let \( ?Bp = C p t - C p s \)
let \( ?Bq = D q t - D q s \)
show \( \text{thesis} \)
proof cases
assume \( ?Bq \leq ?Bp \)
from this ie rb1 rb4 corr-p corr-q bd show \( \text{thesis by simp} \)
next
assume \( \neg ( ?Bq \leq ?Bp ) \)
hence \( ?Bp \leq ?Bq \) by simp
from this ie rb2 rb3 corr-p corr-q bd
have \( |D q t - C p t| \leq |D q s - C p s| + 2*\rho*(t - s) \)
by simp
from this show \( \text{thesis by (simp add: abs-minus-commute)} \)
qed

Drift rate of logical clocks

lemma IC-rate1:
 rho-bound1 \( \lambda p t. IC p i t \)
proof–
{ 
fix \( p::\text{process} \)
fix \( s::\text{time} \)
fix \( t::\text{time} \)
assume cp: correct p t
assume ie: \( s \leq t \)
from cp ie correct-closed have cps: correct p s
by blast
have \( IC p i t - IC p i s \leq (t - s)*(1+\rho) \)
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-1 have PC p t - PC p s ≤ (t - s) * (1 + g) 
by simp
ultimately show ?thesis by simp
qed

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

lemma IC-rate2:
 rho-bound2 (λ p t. IC p i t)
proof -
{
  fix p::process
  fix s::time
  fix t::time
  assume cp: correct p t
  assume ie: s ≤ t
  from cp ie correct-closed have cps: correct p s 
  by blast
  have (t - s) * (1 - ϱ) ≤ 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) * (1 - ϱ) ≤ 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 ⇒ (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 * g * (t - s) 
proof -
  let ?C = ICf i

8
let \( ?D = ICf \ j \)
let \( ?G = |?C \ p \ t - ?D \ q \ t| \leq |?C \ p \ s - ?D \ q \ s| + 2*\varrho*(t - s) \)

from IC-rate1 have rb1: rho-bound1 (ICf \ i \) \& rho-bound1 (ICf \ j \)
  by (simp add: ICf-def)
from IC-rate2 have rb2: rho-bound2 (ICf \ i \) \& rho-bound2 (ICf \ j \)
  by (simp add: ICf-def)

from ie rb1 rb2 corr-p corr-q bounded-drift
have \(?G\) by simp
from this show \(?thesis\) by (simp add: ICf-def)
qed

lemma event-bound:
assumes ie1: \( 0 \leq (t::real) \)
and corr-p: correct \( p \ t \)
and corr-q: correct \( q \ t \)
shows \( \exists \ i. \ t < \max (te \ p \ i) (te \ q \ i) \)
proof (rule contr)
  assume A: \( \neg (\exists \ i. \ t < \max (te \ p \ i) (te \ q \ i)) \)
  show False
  proof
    have F1: \( \forall \ i. \ te \ p \ i \leq t \)
    proof
      fix \( \ i :: nat \)
      from A have \( \neg (t < \max (te \ p \ i) (te \ q \ i)) \)
      by simp
      hence Eq1: \( \max (te \ p \ i) (te \ q \ i) \leq t \) by arith
      have Eq2: \( te \ p \ i \leq \max (te \ p \ i) (te \ q \ i) \)
      by (simp add: max-def)
      from Eq1 Eq2 show \( te \ p \ i \leq t \) by simp
    qed
    have F2: \( \forall (i :: nat). \ correct \ p (te \ p \ i) \)
    proof
      fix \( \ i :: nat \)
      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: \( \forall (i :: nat). \ real \ i * \ rmin \leq te \ p \ i \)
    proof
      fix \( \ i :: nat \)
      show real \( i * \ rmin \leq te \ p \ i \)
      proof (induct \( i \))
        from synch0 show \( real (0::nat) * \ rmin \leq te \ p \ 0 \) by simp
      next
        fix \( \ i :: nat \) assume ind-hyp: \( real \ i * \ rmin \leq te \ p \ i \)
        show real \( (Suc \ i) * \ rmin \leq te \ p \ (Suc \ i) \)
      qed
    qed

from this have rb1: rho-bound1 (ICf \ i \) \& rho-bound1 (ICf \ j \)
from IC-rate1 have rb2: rho-bound2 (ICf \ i \) \& rho-bound2 (ICf \ j \)
from ie rb1 rb2 corr-p corr-q bounded-drift
have \(?G\) by simp
from this show \(?thesis\) by (simp add: ICf-def)
qed
proof –

have Eq1: \(\text{real } i \ast rmin + rmin = (\text{real } i + 1) \ast rmin\)
  by (simp add: distrib-right)
have Eq2: \(\text{real } i + 1 = \text{real } (i+1)\) by simp
from Eq1 Eq2
have Eq3: \(\text{real } i \ast rmin + rmin = \text{real } (i+1) \ast rmin\)
  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 \(\leq\) te p (i+1) − te p i
  by simp
hence Eq4: \(\text{te p } i + rmin \leq \text{te p } (i+1)\) by simp
from ind-hyp have real i \ast rmin + rmin \(\leq\) te p i + rmin
  by (simp)
from this Eq4 have real i \ast rmin + rmin \(\leq\) te p (i+1)
  by simp
from this Eq3 show thesis by simp
qed
qed

have F4: \(\forall (i::nat). \text{real } i \ast rmin \leq t\)
proof
  fix i::nat
  from F1 have te p i \(\leq\) t by simp
moreover
  from F3 have real i \ast rmin \(\leq\) te p i by simp
ultimately show real i \ast rmin \(\leq\) t by simp
qed

from constants-ax have 0 < rmin by simp

from this reals-Archimedean3
have Archi: \(\exists (k::nat). t < \text{real } k \ast rmin\)
  by blast

from Archi obtain k::nat where C: \(t < \text{real } k \ast rmin\)

from F4 have real k \ast rmin \(\leq\) t by simp
hence notC: \(\neg (t < \text{real } k \ast rmin)\) by simp

from C notC show False by simp
qed
qed

2.3 Agreement property

definition \(\gamma 1 x = \pi (2q*\beta + 2*\Lambda) \ (2*\Lambda + x + 2q*(rmax + \beta))\)
definition \(\gamma 2 x = \pi + 2q*rmax\)
definition $\gamma 3 x = \alpha (2*\Lambda + x + 2*\rho*(r_{\text{max}} + \beta)) + \Lambda + 2*\rho*\beta$

definition
okmaxsync :: [nat, Clocktime] ⇒ bool where
okmaxsync i x ←→ (\forall 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 ←→ (\forall 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 okClocks-sym:
assumes ok-pq: okClocks p q i
shows okClocks q p i
proof –
{ fix t :: time
  assume ie1: 0 ≤ t
  assume ie2: t < max (te q i) (te p i)
  assume corr-q: correct q t
  assume corr-p: correct p t

  have max (te q i) (te p i) = max (te p i) (te q i)
  by (simp add: max-def)
  from this ok-pq ie1 ie2 corr-p corr-q
  have |VC q t - VC p t| ≤ δ
  by (simp add: abs-minus-commute okClocks-def)
}
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 (ϕ p (i+1))
using corr-p IClock by (simp add: Adj-def)

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. ϕ q (i+1) n + (PC q (te p (i+1)) - PC q (te q (i+1))))
(is ?T1 = ?T2)
proof –
let ?X = PC q (te p (i+1)) - PC q (te q (i+1))
from corr-q ie1 PC-monotone have posX: 0 ≤ ?X
  by (simp add: le-diff-eq)
from IClock corr-q have \( T1 = \text{cfn} q (\theta q (i+1)) + ?X \)
by(simp add: Adj-def)

from this posX trans-inv show \( \text{thesis} \) by simp
qed

lemma beta-rho:
assumes ie: \( \text{te} q (i+1) \leq \text{te} p (i+1) \)
and corr-p: correct p (\( \text{te} p (i+1) \))
and corr-q: correct q (\( \text{te} p (i+1) \))
and corr-l: correct l (\( \text{te} p (i+1) \))
shows \( |(PC l (\text{te} p (i+1))) - PC l (\text{te} q (i+1))) - (\text{te} p (i+1) - \text{te} q (i+1))| \leq \beta \cdot \varrho \)
proof –
let \( ?X = (PC l (\text{te} p (i+1))) - PC l (\text{te} q (i+1))) \)
let \( ?D = \text{te} p (i+1) - \text{te} q (i+1) \)

from ie have posX: \( 0 \leq ?D \) by simp

from ie PC-monotone corr-l have posX: \( 0 \leq ?X \)
by (simp add: le-diff-eq)
from ie corr-l rate-I have bound1: \( ?X \leq \beta \cdot (1 + \varrho) \) by simp
from ie corr-l correct-closed have corr-l-tq: correct l (\( \text{te} q (i+1) \))
by (blast)
from ie corr-q correct-closed have corr-q-tq: correct q (\( \text{te} q (i+1) \))
by blast
from corr-q-tq corr-p rate-2 have \( |?D| \leq \beta \)
by (simp add: abs-if)
from this constants-ax posD have D-beta: \( \beta \cdot \varrho \leq \beta \cdot \varrho \)
by (simp add: abs-if)

show \( \text{thesis} \)

proof cases
assume A: \( ?D \leq ?X \)
from posX posD A have absEq: \( |?X - ?D| = ?X - ?D \)
by (simp add: abs-if)
from bound1 have bound2: \( ?X - ?D \leq \beta \cdot \varrho \)
by (simp add: mult.commute distrib-right)
from D-beta absEq bound2 show \( \text{thesis} \) by simp
next
assume notA: \( \neg (?D \leq ?X) \)
from this have absEq2: \( |?X - ?D| = ?D - ?X \)
by (simp add: abs-if)
from ie corr-l rate-2 have bound3: \( \beta \cdot (1 - \varrho) \leq ?X \) by simp
from this have \( ?D - ?X \leq \beta \cdot \varrho \) by (simp add: algebra-simps)
from this absEq2 D-beta show \( \text{thesis} \) by simp
qed

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

lemma pe-cond1:
assumes ie: \( \text{te} q (i+1) \leq \text{te} p (i+1) \)
and corr-p: correct p (\( \text{te} p (i+1) \))
and corr-q: correct q (\( \text{te} p (i + 1) \))
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* q * β + 2*Λ
(is ?M ≤ ?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 = θ p (i+1) l − θ q (i+1) l
let ?RE1 = θ p (i+1) l − IC l i (te p (i+1))
let ?RE2 = θ 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 XlD: |?Xl − ?D| ≤ β*q
  by simp

from corr-p corr-q ie beta-rho
have XqD: |?Xq − ?D| ≤ β*q by simp

have TD: |?T − ?D| ≤ 2*Λ + β*q
proof -
    by (simp add: abs-if)
    by(simp add: abs-if)
  have Eq3: |?T − ?ICT| ≤ |?RE1| + |?RE2|
    by(simp add: abs-if)

from readerror corr-p corr-l
have Eq4: |?RE1| ≤ Λ by simp

from corr-l-tq corr-q-tq this readerror
have Eq5: |?RE2| ≤ Λ by simp

from Eq3 Eq4 Eq5 have Eq6: |?T − ?ICT| ≤ 2*Λ
  by simp

have Eq7: ?ICT − ?D = ?Xl − ?D
proof−
from corr-p rte have te p i ≤ te p (i+1)
  by (simp)
from this corr-l correct-closed have corr-l-tpi: correct l (te p i)
  by blast
from corr-q-tq rte have te q i ≤ te q (i+1)
  by simp
from this corr-l-tq correct-closed have corr-l-tqi: correct l (te q i)
  by blast

from IClock corr-l
have F1: IC l i (te p (i+1)) = PC l (te p (i+1)) + Adj l i
  by (simp)
from IClock corr-l-tq
have F2: IC l i (te q (i+1)) = PC l (te q (i+1)) + Adj l i
  by simp
from F1 F2 show ?thesis by (simp)
qed

lemma pe-cond2:
assumes ic: 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*Λ + δS + 2*ϱ*(rmax + β)
proof−
  let ?X = ϑ k (i+1) l − ϑ k (i+1) m
  let ?N = 2*Λ + δS + 2*ϱ*(rmax + β)
  let ?D1 = ϑ k (i+1) l − IC l i (te k (i+1))
  let ?D2 = ϑ k (i+1) m − IC m i (te k (i+1))
  let ?ICS = IC l i (te k (i+1)) − IC m i (te k (i+1))
  let ?tlm = te l i
  let ?IC = IC l i ?tlm − IC m i ?tlm

  have Eq1: |?X| = |(?D1 − ?D2) + ?ICS| (is |E1 = E2)
    by (simp add: abs-if)

  have Eq2: |E2| ≤ |?D1 − ?D2| + |?ICS| by (simp add: abs-if)
from corr-l-tk corr-k beta-bound1 have ie-lk: \( te \ i \leq te \ k \ (i+1) \) 
by (simp add: le-diff-eq)

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

from ie-lk corr-l-tk corr-m-tk IC-bd have Eq3: \(|?ICS| \leq |?IC| + 2g(te k (i+1) - ?tlm)\) 
by simp
from this ind-hyp have Eq4: \(|?ICS| \leq \delta S + 2g(te k (i+1) - ?tlm)\) 
by simp

from corr-l corr-k beta-bound2 have \( te \ k \ (i+1) - ?tlm \leq rmax + \beta \) 
by simp
from this constants-ax have \( 2g*(te k (i+1) - ?tlm) \leq 2g*(rmax + \beta) \) 
by (simp add: real-mult-le-cancel-iff2)
from this Eq4 have Eq4a: \(|?ICS| \leq \delta S + 2g*(rmax + \beta)\) 
by simp

from corr-k corr-l-tk readerror have Eq5: \(|?D1| \leq \Lambda\) 
by simp
from corr-k corr-m-tk readerror have Eq6: \(|?D2| \leq \Lambda \) 
by simp
from this Eq5 Eq6 have Eq7: \(|?D1 - ?D2| \leq 2\Lambda\) 
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 \( l \ i \ (\text{max} \ (te \ l \ i)) - IC \ m \ i \ (\text{max} \ (te \ l \ i)) \) |
\[ \leq \delta S \]
shows \(|\vartheta \ p \ (i+1) \ l - \vartheta \ p \ (i+1) \ m| \)
\[ \leq 2\Lambda + \delta S + 2g*(rmax + \beta) \]
proof–
from corr-p corr-l beta-bound1 have tli-le-tp: \( te \ l \ i \leq te \ p \ (i+1) \) 
by (simp add: le-diff-eq)
from corr-p corr-m beta-bound1 have tmi-le-tp: \( te \ m \ i \leq te \ p \ (i+1) \) 
by (simp add: le-diff-eq)

let \(?tml = \text{max} \ (te \ l \ i) \ (te \ m \ i)\)
from tli-le-tp tmi-le-tp have tml-le-tp: \(?tml \leq te \ p \ (i+1)\) 
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 \( \delta Y = 2 * \Lambda + \delta S + 2 * q(r_{max} + \beta) \)

show \(|\vartheta p (i+1) l - \vartheta p (i+1) m| \leq \delta Y\)

proof cases
  assume A: \( te m i < te l i \)
  from this IC-bound
  have \(|IC l i (te l i) - IC m i (te l i)| \leq \delta S\)
    by (simp add: max-def)
  from this A corr-p corr-l corr-m pe-cond2
  show \(?thesis by (simp)\)
  next
  assume \( \neg (te m i < te l i) \)
  hence Eq1: \( te l i \leq te m i \)
    by simp
  from this IC-bound
  have Eq2: \(|IC l i (te m i) - IC m i (te m i)| \leq \delta S\)
    by (simp add: max-def)
  hence \(|IC m i (te m i) - IC l i (te m i)| \leq \delta S\)
    by (simp add: abs-minus-commute)
  from this Eq1 corr-p corr-l corr-m pe-cond2
  have \(|\vartheta p (i+1) m - \vartheta p (i+1) l| \leq \delta Y\)
    by (simp)
  thus \(?thesis by (simp add: abs-minus-commute)\)

qed

lemma four-one-ind-half:
  assumes ie1: \( \beta \leq r_{min} \)
  and ie2: \( \mu \leq \delta S \)
  and ie3: \( \gamma1 \delta S \leq \delta S \)
  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
  let \(?tpq = te p (i+1)\)

  let \(?f = \lambda n . \vartheta q (i+1) n + (PC q (te p (i+1)) - PC q (te q (i+1)))\)
  let \(?g = \vartheta p (i+1)\)

  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

qed
from Eq1 Eq2 show thesis by simp add: abs-if
qed

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

let ?X = 2*g*\beta + 2*\Lambda
have \forall l. ?ppred l \implies |?f l - ?g l| \leq ?X
proof -
{ 
  fix l
  assume ?ppred l
  from ie4 corr-p corr-q this pe-cond1
  have |?f l - ?g l| \leq (2*g*\beta + 2*\Lambda)
  by (auto)
}
thus thesis by blast
qed

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

let ?Y = 2*\Lambda + \delta S + 2*g*(rmax + \beta)

have \forall l m. ?ppred l \land ?ppred m \implies |?f l - ?f m| \leq ?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 \leq te p (i+1)
  by (simp add: le-diff-eq)
  from corr-p corr-m beta-bound1 have tmi-le-tp: te m i \leq 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 \leq 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| \leq \delta S
  by(auto simp add: okmaxsync-def)
have EqAbs3: \(|f_l - f_m| = |\vartheta q (i+1) l - \vartheta q (i+1) m|\)
    by (simp add: abs-if)

from EqAbs1 corr-q-tq corr-l-tq corr-m-tq theta-bound
have \(|\vartheta q (i+1) l - \vartheta q (i+1) m| \leq \gamma Y\)
    by simp
from this EqAbs3 have \(|f_l - f_m| \leq \gamma Y\)
    by simp

} thus \(?thesis\) by simp
qed

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

have \(\forall \ l \ m. \ ?ppred l \land \ ?ppred m \rightarrow |g_l - g_m| \leq \gamma Y\)
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: \(t l i \leq t p (i+1)\)
      by (simp add: le-diff-eq)
  from corr-p corr-m beta-bound1 have tmi-le-tp: \(t m i \leq t p (i+1)\)
      by (simp add: le-diff-eq)

  let \(?tlm = max (t l i) (t m i)\)
  from tli-le-tp tmi-le-tp have tlm-le-tp: \(?tlm \leq t p (i+1)\)
      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: \(|IC l i ?tlm - IC m i ?tlm| \leq \delta S\)
      by (auto simp add: okmaxsync-def)

  from EqAbs1 corr-p corr-l corr-m theta-bound
  have \(|g_l - g_m| \leq \gamma Y\) by simp
}
  thus \(?thesis\) by simp
qed

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

from correct-count have np - maxfaults \(\leq\) count ?ppred np
    by simp
from this corr-p corr-q cond1 cond2a cond2b prec-enh
have \(|cfn q ?f - cfnp p ?g| \leq \pi \ ?X \ ?Y\)
    by blast

from ie3 this have \(|cfn q ?f - cfnp p ?g| \leq \delta S\)
    by (simp add: \(\gamma 1\)-def)
Theorem 4.1 in Shankar’s paper.

**theorem four-one:**

- **assumes** \( i\text{e}_1: \beta \leq \text{rmin} \)
- **and** \( i\text{e}_2: \mu \leq \delta S \)
- **and** \( i\text{e}_3: \gamma_1 \delta S \leq \delta S \)

**shows** \( \text{okmaxsync } i \delta S \)

**proof**(induct \( i \))

**show** \( \text{okmaxsync } 0 \delta S \)

**proof** -

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

**next**

**fix** \( i \) **assume** ind-hyp: \( \text{okmaxsync } i \delta S \)

**show** \( \text{okmaxsync } (\text{Suc } i) \delta S \)

**proof** -

\[
\begin{align*}
\text{fix } p \ q & \text{ assume corr-p}: \text{correct } p \ (\text{max} \ (\text{te } p \ (i + 1)) \ (\text{te } q \ (i + 1))) \\
\text{assume corr-q}: \text{correct } q \ (\text{max} \ (\text{te } p \ (i + 1)) \ (\text{te } q \ (i + 1)))
\end{align*}
\]
let \( ?tp = te p (i + 1) \)
let \( ?tq = te q (i + 1) \)
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 (te p (i+1))
  by (simp add: max-def)
  from \( A \) corr-q have cq1: correct q (te p (i+1))
  by (simp add: max-def)
  from \( A \)
  have Eq1: \( ?E1 = |IC p (i+1) (te p (i+1)) - IC q (i+1) (te p (i+1))| \)
  (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: \( \neg (\ ?tq < \ ?tp) \)
  from this corr-p have cp2: correct p (te q (i+1))
  by (simp add: max-def)
  from notA corr-q have cq2: correct q (te q (i+1))
  by (simp add: max-def)
  from notA
  have Eq2: \( ?E1 = |IC q (i+1) (te q (i+1)) - IC p (i+1) (te q (i+1))| \)
  (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 (te p (i+1))
  and ie: \( te p (i+1) < te p (i+2) \)
  shows VC p (te p (i+1)) = cfn p (\( \emptyset \) p (i+1))

proof –
  from ie corr-p VClock have VC p (te p (i+1)) = IC p (i+1) (te p (i+1))
  by simp
  moreover
  from corr-p IClock
  have IC p (i+1) (te p (i+1)) = PC p (te p (i+1)) + Adj p (i+1)
  by blast
  moreover
  have PC p (te p (i+1)) + Adj p (i+1) = cfn p (\( \emptyset \) p (i+1))

20
ultimately show \( \theta \) by \( simp \)

**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: \( \gamma_1 \delta S \leq \delta S \)
and ie4: \( \gamma_2 \delta S \leq \delta \)
and ie5: \( \gamma_3 \delta S \leq \delta \)
and ie6: \( te q (i+1) \leq te p (i+1) \)
and ind-hyp: okClocks p q i

**t-bound1**: \( 0 \leq t \)

**t-bound2**: \( t < \max (te p (i+1)) (te q (i+1)) \)

**t-bound3**: \( \max (te p i) (te q i) \leq t \)

**tpq-bound**: \( \max (te p i) (te q i) < \max (te p (i+1)) (te q (i+1)) \)

**corr-p**: correct p t

**corr-q**: correct q t

**shows** |\( VC p t - VC q t \)| \( \leq \delta \)

**proof cases**

**assume** A: \( t < te q (i+1) \)

**let** \( \gamma_{tpq} = \max (te p i) (te q i) \)

**have** Eq1: \( te p i \leq t \land te q i \leq t \)

**proof cases**

**assume** \( te p i \leq te q i \)

**from this** t-bound3 **show** \( \theta \) by (\( simp \ add: \ max-def \))

**next**

**assume** \( \neg (te p i \leq te q i) \)

**from this** t-bound3 **show** \( \theta \) by (\( simp \ add: \ max-def \))

**qed**

**from** ie6 **have** tp-max: \( \max (te p (i+1)) (te q (i+1)) = te p (i+1) \)

**by** (\( simp \ add: \ max-def \))

**from this** t-bound2 **have** Eq2: \( t < te p (i+1) \) 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 \( \gamma_{tpq} \land \gamma_{tpq} \gamma_{tpq} \)

**by** \( blast \)

**from** t-bound3 IC-bd corr-p corr-q

**have** Eq6: \( |IC p i t - IC q i t| \leq |IC p i \gamma_{tpq} - IC q i \gamma_{tpq}| + 2 \ast q*(t - \gamma_{tpq}) (is \ ?E1 \ \leq \ ?E2) \)

**by** \( blast \)

21
from ie1 ie2 ie3 four-one have okmaxsync i \( \delta S \) by simp

from this corr-tpq have \(|IC p i ?tpq - IC q i ?tpq| \leq \delta S\)
by(simp add: okmaxsync-def)

from Eq6 this have Eq7: ?E1 \( \leq \delta S + 2*\rho*(t - ?tpq)\) by simp

from corr-p Eq2 rts0 have \( t - te p i \leq rmax \) by simp
from this have \( 2*\rho*(t - ?tpq) \leq 2*\rho*rmax \)
by (simp add: real-mult-le-cancel-iff1)
hence \( \delta S + 2*\rho*(t - ?tpq) \leq \delta S + 2*\rho*rmax \)
by simp
from this Eq7 have \( ?E1 \leq \delta S + 2*\rho*rmax \) by simp
from this Eq5 ie4 show \( \text{thesis} \) by (simp add: \( \gamma 2\)-def)

next
assume \( \neg (t < te q (i+1)) \)
hence \( B: te q (i+1) \leq t \) by simp

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

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

have \( tq-bound1: t < te q (i+2) \)
proof (rule ccontr)
assume \( \neg (t < te q (i+2)) \)
hence \( C: te q (i+2) \leq t \) by simp

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

have \( te q (i+1) + \beta \leq t \)
proof-
from corr-q-t2 rts1d have \( rmin \leq te q (i+2) - te q (i+1) \)
by simp
from this ie1 have \( \beta \leq te q (i+2) - te q (i+1) \)
by simp
hence \( te q (i+1) + \beta \leq te q (i+2) \) by simp
from this C show \( \text{thesis} \) by simp
qed
from this corr-p corr-q rts2a have \( te p (i+1) \leq t \)
by blast
hence \( \neg (t < te p (i+1)) \) by simp
from this tp-bound2 show False by simp
qed
from tq-bound1 B have tq-bound2: \( te q (i+1) < te q (i+2) \) by simp
from B tp-bound2 have tq-bound3: \( te q (i+1) < te p (i+1) \)
  by simp
from B corr-p correct-closed
have corr-p-tq1: correct p (\( te q (i+1) \)) by blast
from corr-p-tq1 corr-q-tq1 beta-bound1
have tq-bound4: \( te p i \leq \( te q (i+1) \) \)
  by (simp add: le-diff-eq)
from tq-bound1 VClock B corr-q
have Eq1: \( VC q t = IC q (i+1) t \) by simp
from VClock tp-bound1 tp-bound2 corr-p
have Eq2: \( VC p t = IC p i t \) by simp
from Eq1 Eq2 have Eq3: \( |VC p t - VC q t| = |IC p i t - IC q (i+1) t| \)
  by simp
from this Eq3
have IC-split: \( |IC p i t - IC q (i+1) t| \leq \)
  \( |IC p i (\( te q (i+1) \)) - IC q (i+1) (\( te q (i+1) \))| + 2*\( q* (t - te q (i+1)) \) \)
  by simp
from tq-bound2 VClock corr-q-tq1
have Eq4: \( VC q (\( te q (i+1) \)) = IC q (i+1) (\( te q (i+1) \)) \) by simp
from this tq-bound2 VClock corr-q-tq1
have Eq5: \( IC q (i+1) (\( te q (i+1) \)) = cfn q (\( \vartheta q (i+1) \)) \) by simp
hence IC-eq-cfn: \( IC p i (\( te q (i+1) \)) - IC q (i+1) (\( te q (i+1) \)) = \)
  \( IC p i (\( te q (i+1) \)) - cfn q (\( \vartheta q (i+1) \)) \)
(is \( ?E1 = ?E2 \))
  by simp
let \( ?f = \vartheta q (i+1) \)
let \( ?ppred = \lambda l. correct l (\( te q (i+1) \)) \)
let \( ?X = 2*\( q* (r_{max} + \beta) \) \)
have \( \forall l m. ?ppred l \land ?ppred m \longrightarrow \vartheta q (i+1) l - \vartheta q (i+1) m \leq ?X \)
proof-
{
  fix l :: process
  fix m :: process
  assume corr-l: ?ppred l
assume corr-m: ?ppred m

let ?tlm = max (te l i) (te m i)
have tlm-bound: ?tlm ≤ te q (i+1)

proof
  from corr-l corr-q-tq1 beta-bound1 have te l i ≤ te q (i+1)
  by (simp add: le-diff-eq)
moreover
  from corr-m corr-q-tq1 beta-bound1 have te m i ≤ te q (i+1)
  by (simp add: le-diff-eq)
ultimately show ?thesis by simp
qed

from tlm-bound corr-l corr-m correct-closed
have corr-tlm: correct l ?tlm ∧ correct m ?tlm
  by blast

have |IC l i ?tlm − IC m i ?tlm| ≤ δS
proof
  from ie1 ie2 ie3 four-one have okmaxsync i δS
  by simp
  from this corr-tlm show ?thesis by (simp add: okmaxsync-def)
qed

from this corr-l corr-m corr-q-tq1 theta-bound
have |ϑ q (i+1) l − ϑ q (i+1) m| ≤ ?X by simp

thus ?thesis by blast
qed

hence readOK: okRead1 (ϑ q (i+1)) ?X ?ppred
  by (simp add: okRead1-def)

let ?E3 = cfn q (ϑ q (i+1)) − ϑ q (i+1) p
let ?E4 = ϑ q (i+1) p − IC p i (te q (i+1))

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


from correct-count have ppredOK: np − maxfaults ≤ count ?ppred np
  by simp
from readOK ppredOK corr-p-tq1 corr-q-tq1 acc-prsv
have |?E3| ≤ α ?X
  by blast
from this Eq8 have Eq9: |?E2| ≤ α ?X + |?E4| by simp

from corr-p-tq1 corr-q-tq1 readerror
have |?E4| ≤ Λ by simp

from this Eq9 have Eq10: |?E2| ≤ α ?X + Λ by simp

from this VC-split IC-eq-cfn
have almost-right:
  |VC p t − VC q t| ≤
\[
\alpha \ ?X + \Lambda + 2*\rho*(t - te \ q \ (i+1))
\]
by \(\text{simp}\)

have \(t - te \ q \ (i+1) \leq \beta\)

proof (rule ccontr)
  assume \(\neg (t - te \ q \ (i+1) \leq \beta)\)
  hence \(te \ q \ (i+1) + \beta \leq t\) by \(\text{simp}\)
  from this corr-p corr-q rts2a have \(te \ p \ (i+1) \leq t\)
  by \(\text{auto}\)
  hence \(\neg (t < te \ p \ (i+1))\) by \(\text{simp}\)
  from this tp-bound2 show False
  by \(\text{simp}\)
qed

from this constants-ax
have \(\alpha \ ?X + \Lambda + 2*\rho*\beta\)
by \(\text{simp}\)

from this almost-right
have \(|VC \ p \ t - VC \ q \ t| \leq \alpha \ ?X + \Lambda + 2*\rho*\beta\)
by \(\text{arith}\)

from this ie5 show \(\text{thesis}\) by \(\text{simp add: }\gamma3\text{-def}\)
qed

Theorem 4.2 in Shankar’s paper.

\textbf{theorem four-two:}\nassumes \(\text{ie1: } \beta \leq rmin\)
and \(\text{ie2: } \mu \leq \delta S\)
and \(\text{ie3: } \gamma1 \ \delta S \leq \delta S\)
and \(\text{ie4: } \gamma2 \ \delta S \leq \delta\)
and \(\text{ie5: } \gamma3 \ \delta S \leq \delta\)
shows \(\text{okClocks } p \ q \ i\)
proof (induct \(i\))
show \(\text{okClocks } p \ q \ 0\)
proof
  \{
  fix \(t::\text{time}\)
  assume t-bound1: \(0 \leq t\)
  assume t-bound2: \(t < \text{max} \ (te \ p \ 0) \ (te \ q \ 0)\)
  assume corr-p: \(\text{correct } p \ t\)
  assume corr-q: \(\text{correct } q \ t\)
  from t-bound2 synch0 have \(t < 0\)
  by (simp add: max-def)
  from this t-bound1 have False by \(\text{simp}\)
  hence \(|VC \ p \ t - VC \ q \ t| \leq \delta\) by \(\text{simp}\)
  \}
thus \(\text{thesis}\) by (simp add: okClocks-def)
qed
next
fix \(i::\text{nat}\) assume ind-hyp: \(\text{okClocks } p \ q \ i\)
show \(\text{okClocks } p \ q \ (\text{Suc } i)\)
proof –
{
  fix \( t :: \) time
  assume t-bound1: \( 0 \leq t \)
  assume t-bound2: \( t < \max (\text{te } p (i+1)) (\text{te } q (i+1)) \)
  assume corr-p: correct p t
  assume corr-q: correct q t

  let \( \forall tpq1 = \max (\text{te } p i) (\text{te } q i) \)
  let \( \forall tpq2 = \max (\text{te } p (i+1)) (\text{te } q (i+1)) \)

  have \( |\text{VC } p t - \text{VC } q t| \leq \delta \)
  proof
    cases
    assume tpq-bound: \( \forall tpq1 < \forall tpq2 \)
    show \( \forall \text{thesis} \)
      proof
        cases
        assume \( t < \forall tpq1 \)
        from t-bound1 this corr-p corr-q ind-hyp
        show \( \forall \text{thesis} \)
          by simp add: okClocks-def
        next
        assume \( \neg (t < \forall tpq1) \)
        hence \( \forall tpq-le-t: \forall tpq1 \leq t \) by arith

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

def
next
assume \(\neg (\exists t\in \mathbb{Q}) t < t\)
hence ?tpq2 \leq ?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: \(\beta \leq \gamma\)
and ie2: \(\mu \leq \delta\)
and ie3: \(\gamma_1 \delta \leq \delta S\)
and ie4: \(\gamma_2 \delta \leq \delta\)
and ie5: \(\gamma_3 \delta \leq \delta\)
and ie6: \(0 \leq t\)
and cpq: correct \(p, t\) \& correct \(q, t\)
shows \(|VC p t - VC q t| \leq \delta\)

proof
from ie6 cpq event-bound have \(\exists i :: nat. t < max (te p i) (te q i)\)
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

