A Formal Model: Media Access Control and Frame and Symbol Processing

FlexRay Seminar

Peter Böhm, 21.10.2005

Overview

- The Model
  - Architecture
  - Clock Synchronization
  - Schedule
- Main Theorem
- Proof of Theorem

Motivation

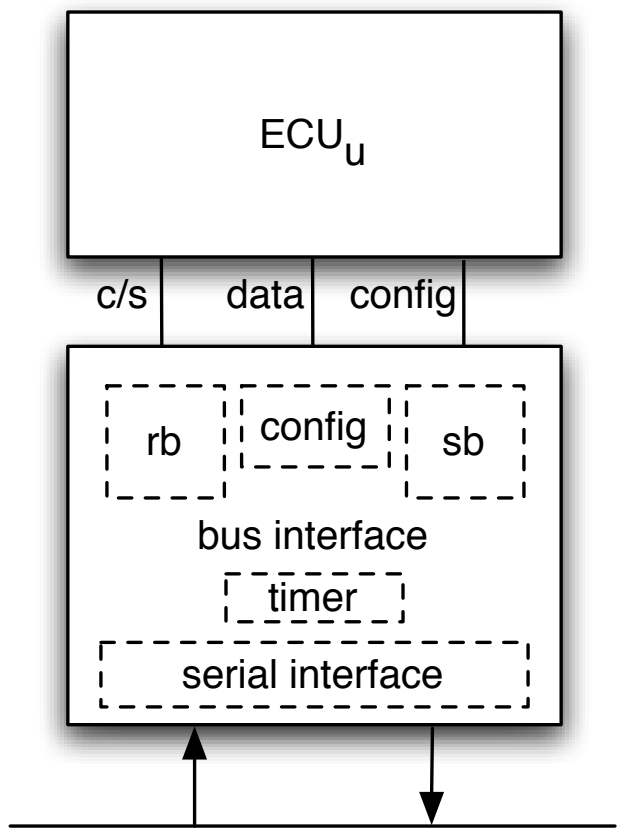
- formal model for FlexRay *like* bus interface
- integration of serial interface into bus interface
- omit all features related to fault-tolerance
- differences within:
  - clock synchronization
  - schedule

Architecture

- bus with *n* electronic control units (ECU):

  ECU_u with $u \in [1:n]$

- ECU connected to bus interface via I/O-ports:
  - control and status port (*c/s*)
  - data port (*data*)
  - configuration port (*config*)
- bus interface:
  - send (*sb*) and receive buffer (*rb*)
  - configuration
  - timer
  - serial interface



Buffers

- accessed via data port
- 2 pointers: *sbp* into *sb*, *rbp* into *rb*
- writing to data port:
  - data to address *sbp* in *sb*
  - increment *sbp*
  - ➡ successive writes fill the send buffer
- reading from data port:
  - read from address *rbp* in *rb*
  - increment *rbp*
  - ➡ successive reads read out receive buffer

Timer

- hardware timer: ti_u^i
- incremented every 8 clock ticks
- correspond to macroticks in FlexRay

  ➡ simplification
- function: $ati_u(t)$

  $ati_u(t) = ti_u^i$ *if* $t \in [e_u(i), e_u(i+1))$

  $e_u(i)$ denotes the i-th rising edge of the local clock
- timers of different interfaces synchronized by the clock synchronization
- local time base for interrupts

Configuration

- written during startup phase via *config port*
- components:
  - *u*: id of the ECU attached to bus interface
  - *S*: global schedule
  - *wakeup*: processor wakeup function
- *wakeup:*
  - processor needs time to access the buffers between transmission times
  - at time *wakeup(σ)* a timer interrupt is activated

Configuration

- *S = (ns, ecu, st, mlen)*
  - *ns*: number of slots: $\sigma \in [0:ns-1]$
  - *ecu*: ECU sending during slot σ specified by $ecu(\sigma) \in [1:n]$
  - st: during slot σ transmission starts at time st(σ)
  - *mlen*: *mlen(σ)* specifies the length of the message transmitted in slot σ
- transmission: from *st(σ)* to *wakeup(σ)*
- processor access: from *wakeup(σ)* to *st(σ+1)*

Clock Synchronization

- different to FlexRay
- simple clock synchronization algorithm
- reset timers after transmission of last message within each round
- sending ECU: timer reset after last FES[0] copy

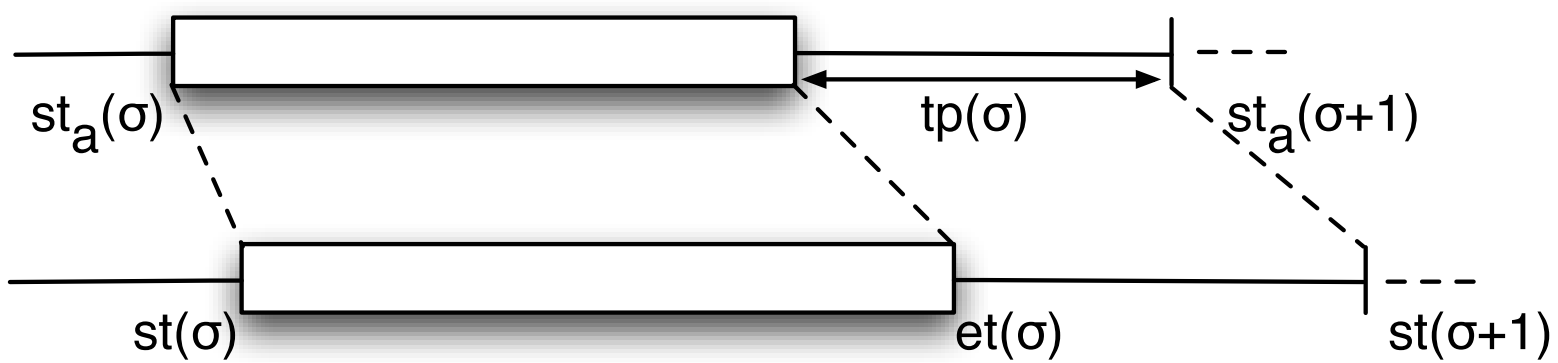
  FES[0]: the last bit of a frame
- receiver: reset timer 3 clock ticks after sampling of FES[0]
- Assumption: clock drift bounded by δ

Schedule

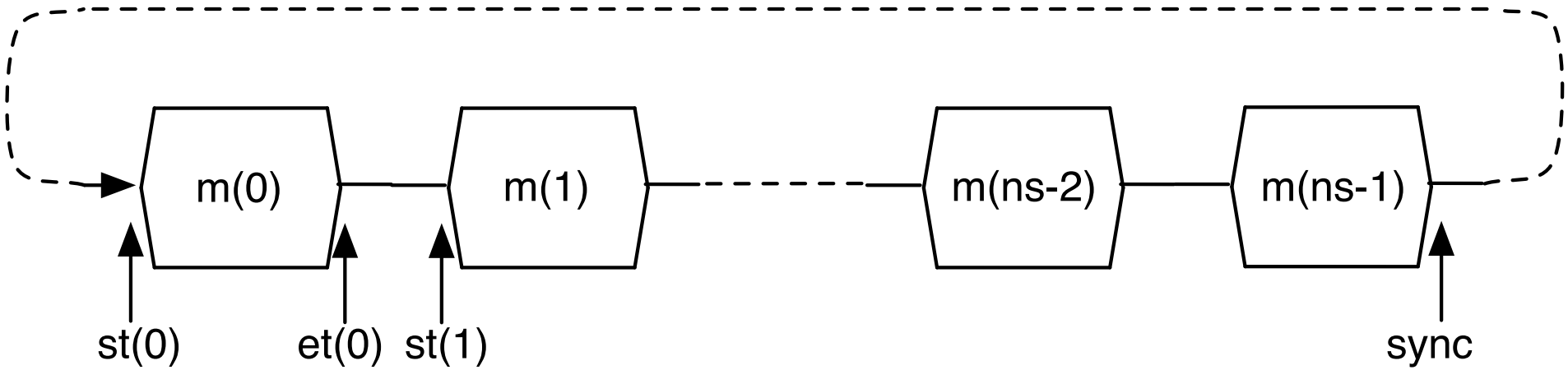
- definition: abstract start time $st_a(\sigma)$
  - start time if there would be no clock drift

    $st_a(0) = 0$

    $st_a(\sigma+1) = st_a(\sigma) + l + tp(\sigma)$ with $l = 10*mlen(\sigma) + 4$
  - $tp(\sigma)$: the timer ticks for ECU to access the serial interface
- start time with clock drift: $st(\sigma) = st_a(\sigma)*(1+\delta)$
- transmission end time: $et(\sigma) = (st(\sigma)+l)*(1+\delta)$



Schedule



Main Theorem

After message transmission, $rb_u = sb_s$ for any ECU u and sending ECU s

proof outline:

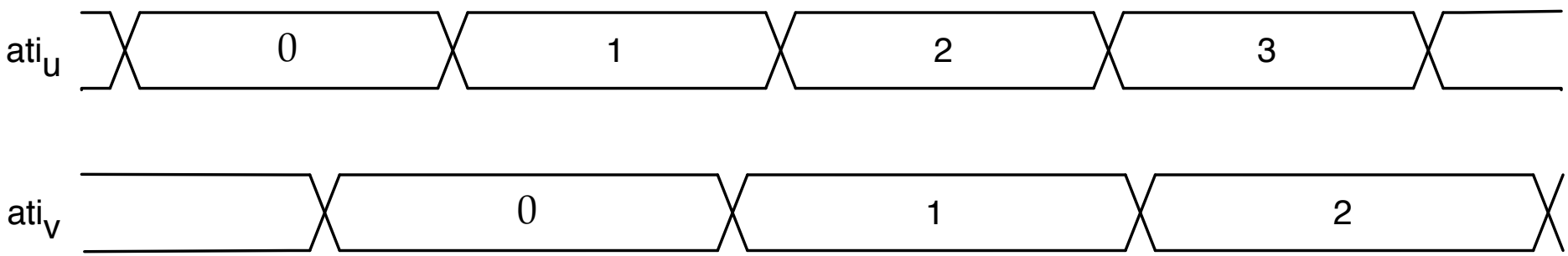
1. proof that timers are bound due to clock synchronization
2. transmission times of different slots do not overlap

Definition

- *time(v;u,T) := min{$ati_v(t)|ati_u(t) = T$}*

  local time on interface *v* at local time *T* on interface *u*

- Example:



  - time(u;v,1) = 1, time(u;v,2) = 3
  - time(v;u,1) = 0, time(v;u,2) = 1

Lemma 1

For all u,v: time(v;u,0) = 0

Proof:

reset of receiver's timer:

*str(k) = cy(8*k) + [5:8] +1*

⇔ *str(k)+3 = cy(8*k) + [9:12]*

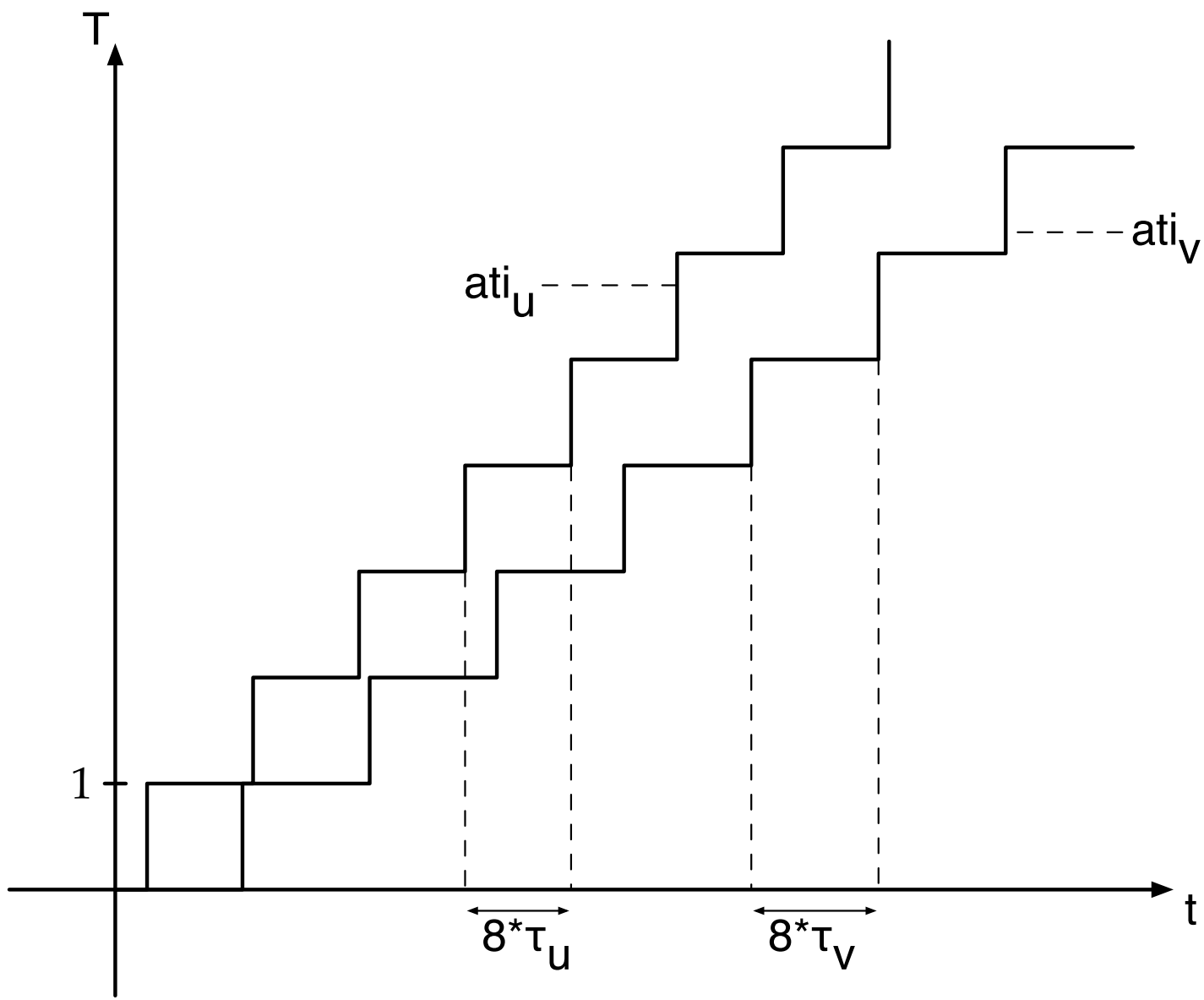
reset of sender's timer:

9 clock ticks after first FES[0] copy

⇒ difference < 8 ⇒ *time(v;u,0) = 0 for all u,v*

Timer Drift

timer drift?

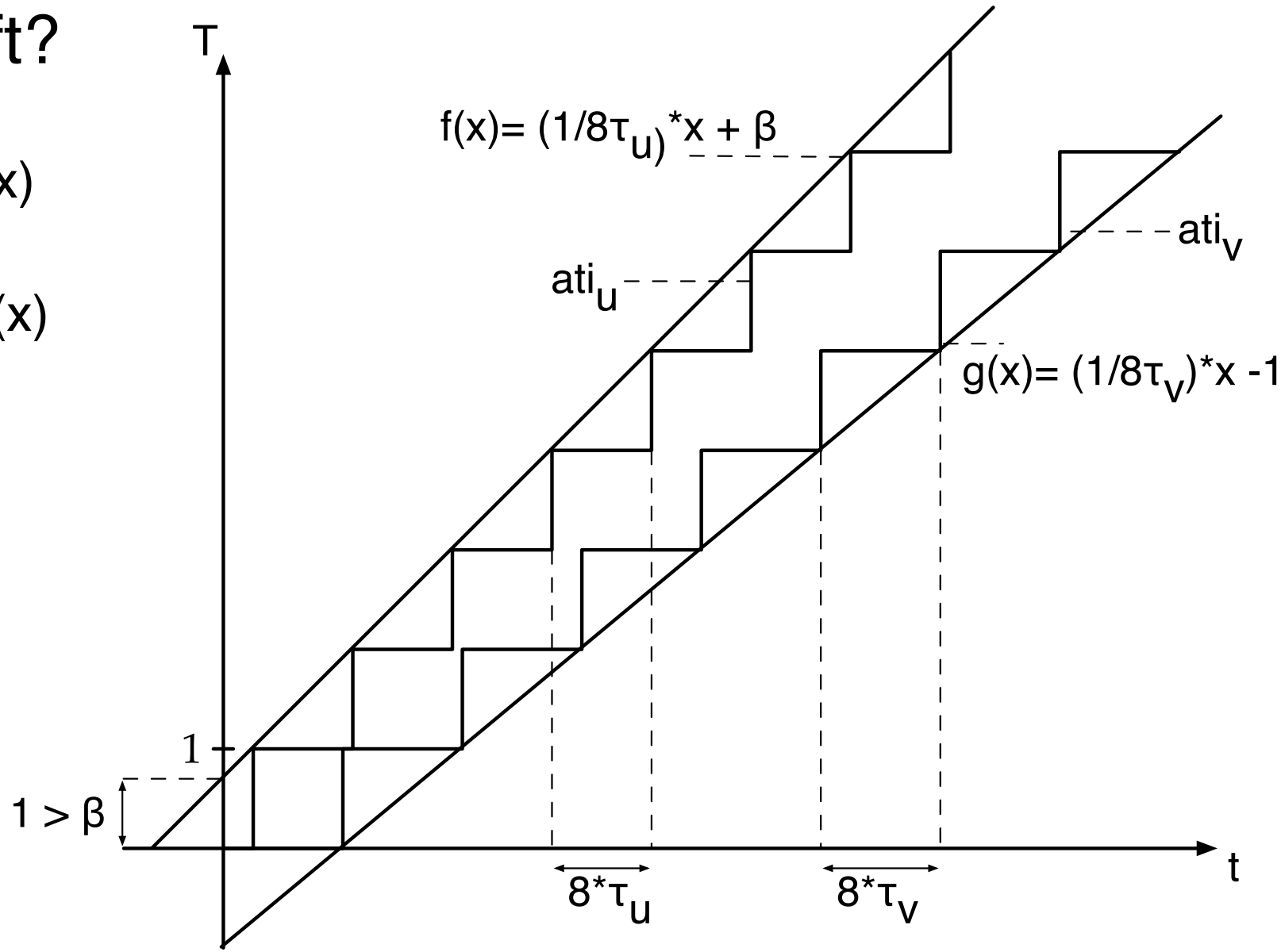


Timer Drift

timer drift?

ati_u upper bound by f(x)

ati_v lower bound by g(x)



Timer Drift

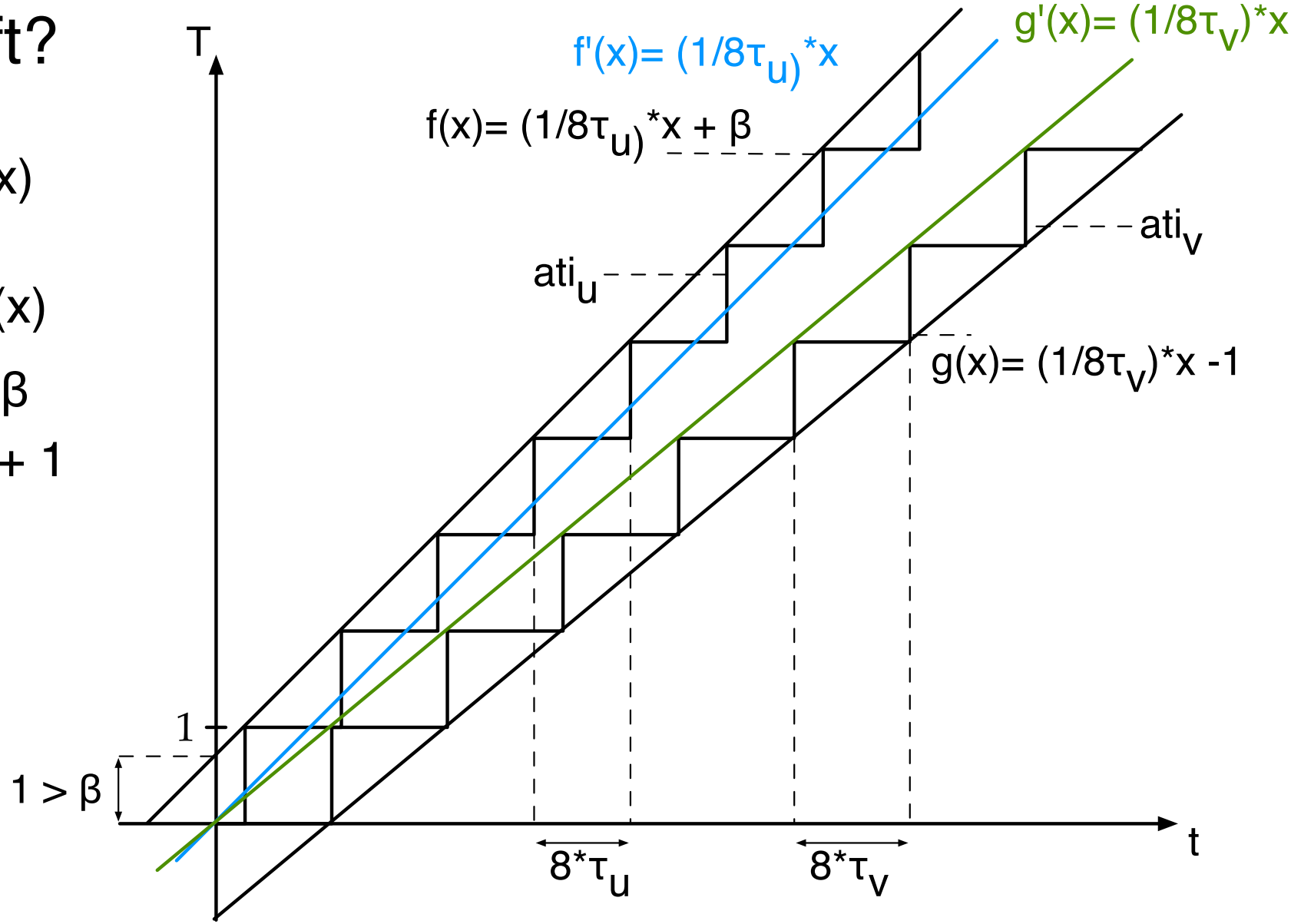
timer drift?

ati_u upper bound by f(x)

ati_v lower bound by g(x)

$f'(x) = f(x) - \beta$

$g'(x) = g(x) + 1$



Timer Drift

timer drift?

ati_u upper bound by f(x)

ati_v lower bound by g(x)

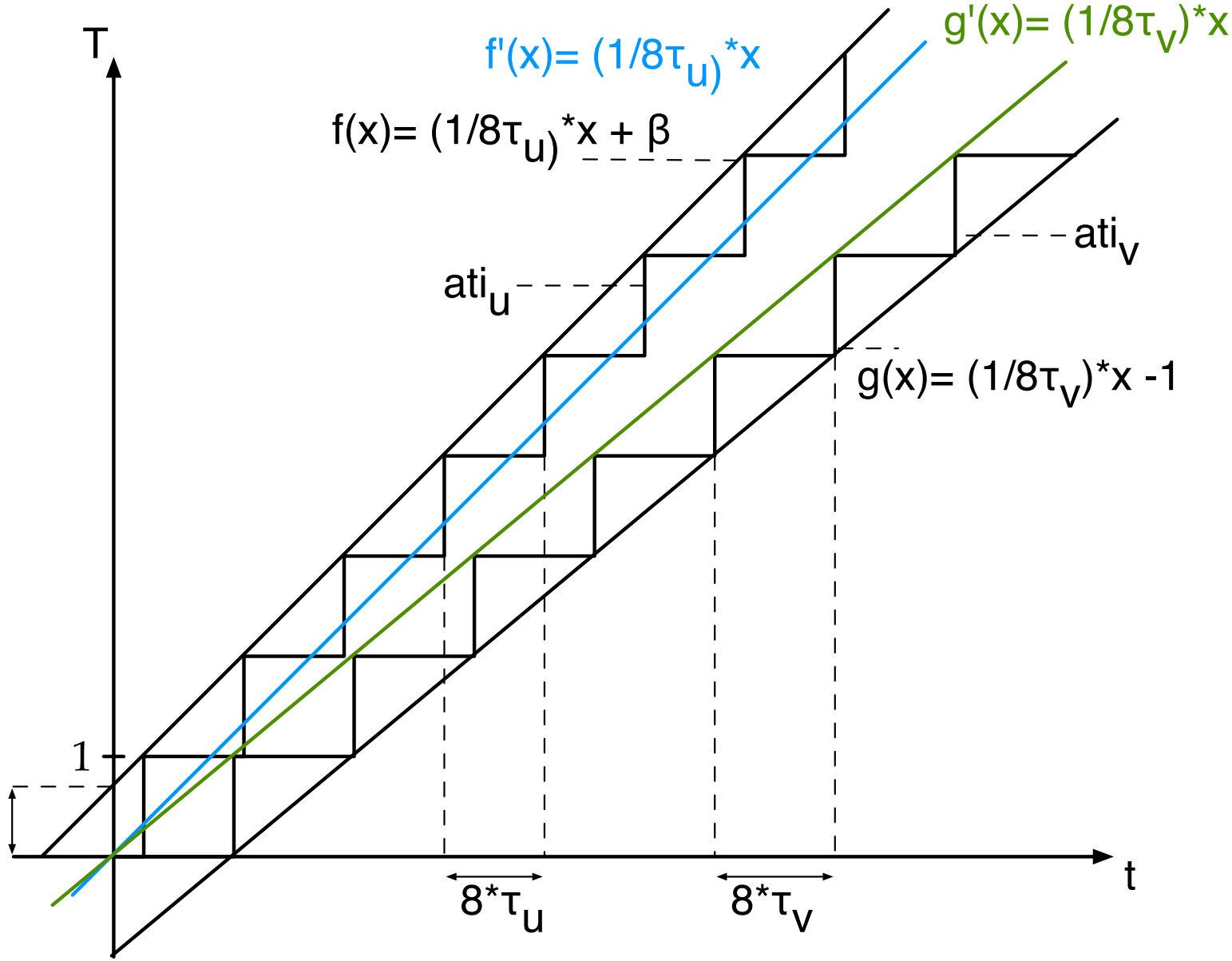
$f'(x) = f(x) - \beta$

$g'(x) = g(x) + 1$

$\beta < 1 \Rightarrow$

$|f(x)-g(x)| \leq |f'(x)-g'(x)|+2$

$|f'(x)-g'(x)| \leq \delta T$



Lemma 2

For all u,v and times T the timer drift is bounded by

$$| time(v;u,T) - T | \leq T*\delta + 2$$

Proof: Follows from previous arguments

Lemma 3

On any ECU *u* the serial interface is idle during

[et(σ)+3:time(u;ecu(σ+1),st(σ+1))]

for sufficient *tp(σ)*

Proof

1. bus idle after *et(σ)+3:*

   from L2 follows $time(u;v,T) \leq (1+\delta)*T+2$

   $\Rightarrow time(u;ecu(\sigma),st(\sigma)+l) \leq (1+\delta)*(st(\sigma)+l)+2 \leq et(\sigma)+2$

   $\Rightarrow$ *et(σ)+3:* serial interface is idle

2. $et(\sigma)+3 \leq time(u;ecu(\sigma+1),st(\sigma+1))$ for sufficient *tp(σ)*

   easy, but long

Lemma 4

For any ECU *u* and any *t* with

$ati_u(t) \in [et(\sigma) + 3: time(u;ecu(\sigma+1),st(\sigma+1)]$

$rb_u^t[0:l`-1] = sb_{ecu(\sigma)}^{8*st(\sigma)}[0:l`-1]$

with $l` = mlen(\sigma)$

Proof: Follows from Lemma 3 and theorem about serial interface

6. Proof of Theorem

- Lemma 2 ⇒ timer drift is bound
- Lemma 3 ⇒ slots do not overlap
- Lemma 4 ⇒ $rb_u = sb_s$ for any u and sender v

⇒ Theorem