A Control-Based Approach to Transport Channel Synchronization in UTRAN

Juan J. Alcaraz, Gaspar Pedreño, and Fernando Cerdan

***Abstract*— Transport channel synchronization in 3G access networks is currently based on a simple algorithm that tries to track the delay of the link by adding or subtracting a constant step. We propose an algorithm that improves the responsiveness of the system under abrupt delay changes, reducing frame losses. Our proposal is faster than the proportional tracking system upon which it is based and has delay-independent stability.**

***Index Terms*— Transport channel synchronization, UTRAN, delays.**

I. Introduction

In 3G wireless networks, link-level communication between Nodes B and RNCs is regulated by the Frame Protocol (FP), specified by the 3GPP in [1][2]. By means of the transport channel synchronization procedure, FP adjusts the sending time of the downlink frames for each data channel, in order that each frame arrives to the Node B within a time reception window of configurable length, w, assuring that each frame is transmitted at its corresponding Transmission Time Interval (TTI) with the less possible buffering at the Node B.

Fig. 1 illustrates this synchronization procedure. The RNC holds an estimation of the downlink delay (offset). When delay variations cause a frame reception outside the window, the Node B responds with a Timing Adjustment (TA) containing the Time Of Arrival (TOA) of this frame. If no TA is received, the RNC sends the upcoming frame in $t_n = t_{n-1} + t_{TTI}$, where t_{TTI} is the duration of a TTI. In the classic adjustment algorithm [3], if $TOA > 0$ (early arrival), then $t_n = t_{n-1} + t_{TTI} + K$, with K constant, which is equivalent to $offset_n = offset_{n-1} - K$, where $offset_n$ is the offset predicted for frame n. If $TOA < 0$, (late arrival), then $t_n = t_{n-1} + t_{TTI} - K$. Note that if a frame arrives so late that the Node B is not able to process it before its corresponding transmission interval on the air interface, this frame is discarded.

The classic algorithm, however, reacts too slowly against abrupt increments of the delay, causing several consecutive frame losses. These situations may arise in UTRAN IP, e.g. because of path restoration mechanisms as in [4].

The main contribution of this work is the use of discrete-time control theory to propose an offset adjustment algorithm that improves the classic one, without requiring any change in 3GPP specifications. Our proposal is detailed in Section II. In Section III we show and compare numerical and simulation-based performance results. Conclusions are offered in Section IV.

Manuscript received March 5, 2007. The associate editor coordinating the review of this letter and approving it for publication was Prof. Carla-Fabiana Chiasserini. This work was partially supported under projects *m: Ciudad* (FIT-330503-2006-2) and *CSI-RHET* (TEC2005-08068-C04-01/TCM).

The authors are with the Dept. of Information Technologies and Communications, Technical University of Cartagena (UPCT), Spain (email: {juan.alcaraz, gaspar.pedreno, fernando.cerdan}@upct.es).

Digital Object Identifier 10.1109/LCOMM.2007.070331.

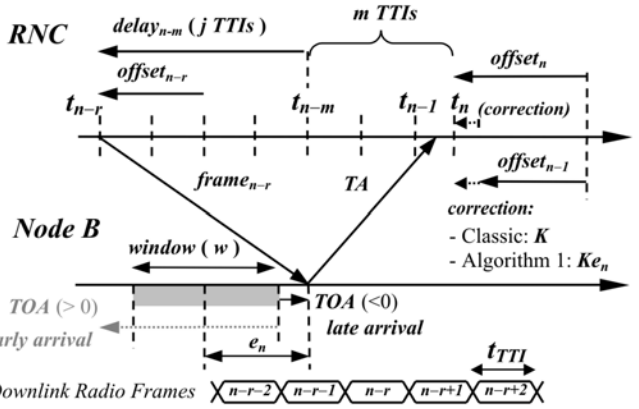

Fig. 1. Reception of downlink frame $n - r$ and subsequent correction on frame n.

II. Proposed Algorithm

Because the offset at the RNC has to track the downlink delay, a feasible adjustment algorithm, based on control theory, is the proportional tracking system (algorithm 1). Considering the centre of the window as the optimum arrival time, the error signal of the control system for frame n is $e_n = w/2 - TOA$ (see Fig. 1), resulting in $offset_n = offset_{n-1} + Ke_n$ where K is the gain of the system. Defining the Round Trip Time (RTT) as the time elapsed between the sending time of a frame and the moment when the TA feedback for this frame is applied to another frame, the number of TTIs in the RTT is $r = \lceil RTT/t_{TTI} \rceil = j + m$, where j and m are the downlink and uplink delay in TTIs respectively. Therefore, $e_n = delay_{n-m} - offset_{n-r}$, where $delay_{n-m}$ is the delay perceived by the Node B at time-slot $n - m$. Expressing algorithm 1 as a difference equation,

$$x(n) = x(n-1) + K(u(n-m) - x(n-r)) \qquad (1)$$

where $x(n)$ is $offset_n$ and $u(n)$ is $delay_n$. Our proposal is a modification of algorithm 1, based on applying the correction to the estimation in which the error is measured. This requires two changes. First, to compare the delay measured at time-slot $n - m$ with the offset applied to frame $n - m$, i.e. the new error signal is $e'(n) = u(n-m) - x(n-m)$. Second, to make the adjustment on $x(n-m)$ instead of $x(n-1)$, because the system does not have any information about the error concerning $x(n-1)$. Therefore,

$$x(n) = x(n-m) + K(u(n-m) - x(n-m)) \qquad (2)$$

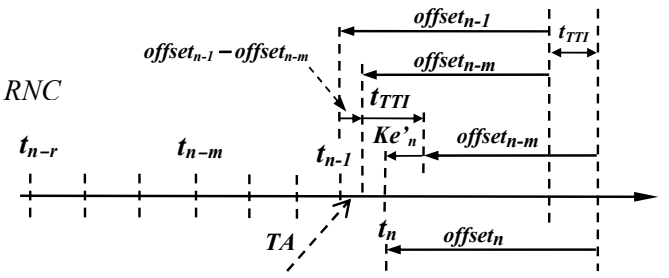

Fig. 2. Time diagram for algorithm 2.

In order to implement algorithm 2 from the actual e_n value present at the RNC, we develop (2) as

$$x(n) = x(n-m)+K(u(n-m)-x(n-r)+x(n-r)-x(n-m)),$$

which equals to

$$x(n) = x(n-m) + K(e(n) + x(n-r) - x(n-m)) \quad (3)$$

The first $x(n-m)$ term implies a shift in the time instant upon which the next sending time is computed. Thus we can express (3) as

$$t_n = t_{n-1} + offset_{n-1} - offset_{n-m} + t_{TTI} - K(e_n + offset_{n-r} - offset_{n-m}), \quad (4)$$

Fig. 2 illustrates the operation of algorithm 2 for $offset_{n-m} < offset_{n-1}$, and $e_n > 0$. The frame at t_{n-1} is transmitted with $offset_{n-1}$. Because the frame at t_n is to be transmitted with $offset_{n-m} + Ke'_n$, where $e'_n = e_n + offset_{n-r} - offsset_{n-m}$, we must relocate the reference time to $t_{n-1} + offset_{n-1} - offset_{n-m}$. Finally, we obtain the next sending time, t_n, by adding $t_{TTI} - Ke'_n$ to this reference time. Obviously, the FP entity at the RNC must keep the last r offset values. Likewise, the algorithm must update its estimation of j and m, say $j \approx \lceil offset/t_{TTI} \rceil$ and $m = r - j$, where r can be periodically sampled by the node synchronization procedure described in [2].

A. Analysis and Gain Design

Making the z-transform of (2), and denoting $X(z)$ and $U(z)$ as the z-transforms of $x(n)$ and $u(n)$ respectively we get the transfer function of a system implementing algorithm 2:

$$\frac{X(z)}{U(z)} = \frac{K}{z^m - 1 + K}. \quad (5)$$

Algorithms 1 and 2 can be analyzed as discrete-time control systems making certain assumptions. First, the RNC has the TOA of every frame, i.e. $w = 0$. Second, there are no idle times for data traffic in the period under study. Third, the inter-departure time between frames at the RNC is constant and equal to t_{TTI}. In classical control design, the gain K is determined from a specified damping ratio (ζ), related to the overshoot of the step response. In addition, the stability of the system requires $\zeta > 0$. According to ref. [5], for $\zeta < 1$, the roots of the closed-loop poles are related to ζ with

$$z = e^{-\frac{2\pi\zeta}{\sqrt{1-\zeta^2}}\omega} e^{j2\pi\omega}, \quad (6)$$

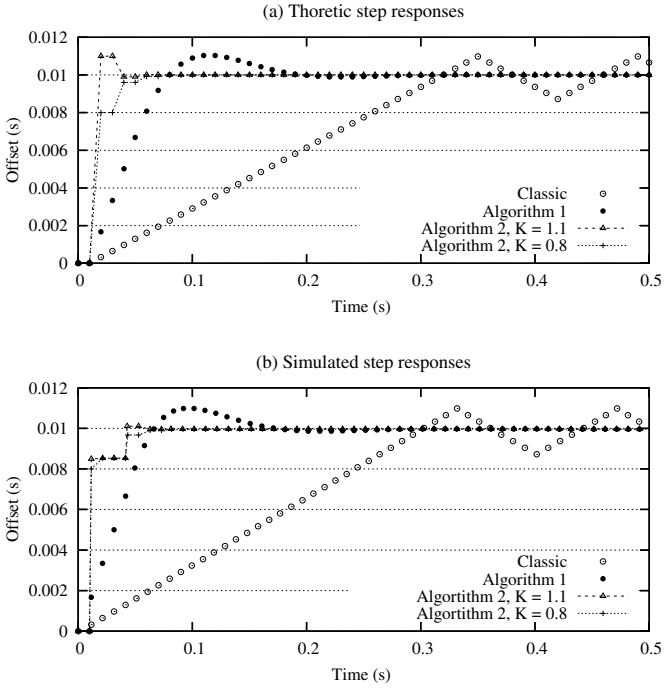

Fig. 3. Theoretical and simulated step responses (w = 0).

where $\omega = \omega_d/\omega_s$, ω_d is the transient oscillation frequency and ω_s is the sampling frequency in rad/s. In our case $\omega_s = 2\pi/t_{TTI}$. At the same time, by (5) the closed-loop poles satisfy $z^m = 1 - K$, which applied in (6) gives:

$$e^{-\frac{2\pi\zeta}{\sqrt{1-\zeta^2}}\omega m} e^{j2\pi\omega m} = 1 - K. \quad (7)$$

Because K is a real number:

$$e^{-\frac{2\pi\zeta}{\sqrt{1-\zeta^2}}\omega m} sin(2\pi\omega m) = 0.$$

The above equation has roots at $\omega = 0$, which corresponds to the trivial case $K = 0$, and at $\omega = n/2m$, for $n = 1, 2, \ldots, m-1$. The root for $n = 1$ gives ω of the dominant pole, as can be easily checked representing the root locus of the system. Using this ω in (7) we get

$$K = 1 + e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}. \quad (8)$$

The above value does not depend on the grade of the characteristic equation, m. In order to illustrate the benefit of this fact, consider the algorithm 1 with $\zeta = 0.6$. In this case, if $r = 2$, then $K = 0.385$, but if $r = 4$, the gain must be set to $K = 0.165$. In contrast, in algorithm 2, if we set $K = 1.1$, the system remains with $\zeta = 0.6$ regardless of the delay. Moreover, algorithm 1 is unstable for $K > 1$ if $r = 2$ and for $K > 0.44$ if $r = 4$, while algorithm 2 is always stable if $K < 2$. Consequently, algorithm 1 must keep track of the RTT and recalculate K accordingly for a stable operation.

III. Validation and Performance Results

The topology of the simulated scenario consists of a point-to-point transmission path, set to 10 Mbit/s between the RNC and the Node B. We consider a single standard data channel of 384 Kbit/s with $t_{TTI} = 10\,ms$.

Fig. 3 (a) presents the theoretical step responses for algorithms 1, 2 and the classic one, with $r = 4$ and a delay increment $I = 10\ ms$. Algorithm 1 and algorithm 2 with $K = 1.1$ have an overshoot of 10%. Algorithm 2 with $K = 0.8$ is overdamped. The classic algorithm oscillates when the window size is $w = 0$. With $K = 0.32\ ms$, its oscillation amplitude is set to 10% of I. Fig. 3 (b) shows the simulated responses, with similar configuration. The discrepancy at the initial stages of the theoretical and simulated responses of algorithm 2 with $K = 1.1$ happens because in practice the sending time advance cannot be greater than t_{TTI} (see Fig. 2). However, even overdamped algorithm 2 raises and stabilizes faster than algorithm 1. If $w > 0$, the feedback with TA frames stops when $e_n < w/2$. Therefore the final estimation is less accurate, but can be reached faster and without oscillations, provided ζ is high enough and r is not too large. The rise time, t_r, is the time it takes the system to achieve the steady-state value for the first time. Considering the system with parameters $w = 5\ ms$, $I = 10\ ms$ and $r = 4$, algorithm 1 ($\zeta = 0.6$) has $t_r = 86\ ms$ and algorithm 2 ($K = 0.8$) has $t_r = 71\ ms$. If the Iub link has a larger delay, e.g. $r = 6$, then $t_r = 146\ ms$ for algorithm 1 and $t_r = 91\ ms$ for algorithm 2.

IV. Conclusions

We have used discrete-time control theory to develop a mechanism (algorithm 2) for transport channel synchronization at the Iub. Algorithm 2 is an enhanced version of a proportional tracking system (algorithm 1). While, compared to the classic one, both algorithms improve the step response, algorithm 2 is faster than algorithm 1 because algorithm 2 has fewer poles in its transfer function and can operate with higher gain without losing stability, independently of the delay. In consequence, algorithm 2 does not need to reconfigure its gain under delay variations.

Acknowledgment

The authors would like to thank Prof. Joan Garcia-Haro for his valuable comments on this letter.

References

[1] 3rd Generation Partnership Project; Technical Specification Group RAN, "Synchronisation in UTRAN Stage 2, 7.2.0, 2006-12," 3GPP TS 25.402, Dec. 2006.

[2] 3rd Generation Partnership Project; Technical Specification Group RAN, "Iub/Iur Interface User Plane Protocol for DCH Data Streams, 7.3.0, 2006-12," 3GPP TS 25.427, Dec 2006.

[3] M. Sagfors, J. Peisa and S. Malomsky, "Transmission offset adaptation in UTRAN," in *Proc. IEEE VTC*, vol. 7, pp. 5240-44, Sept. 2004.

[4] T. Bu, M. C. Chan, and R. Ramjee, "Connectivity, performance, and resiliency of IP-based CDMA radio access networks," *IEEE Trans. Mobile Computing*, vol. 5, no. 8, pp. 1103-18, Aug. 2006.

[5] K. Ogata, *Discrete Time Control Systems*, 2nd Ed. Upple Saddle River, NJ: Prentice Hall Inc., 1995, pp. 173-224.