3GPP TSG RAN WG1#9 ***TSGR1#9(99)j81***
November 30 – December 3, 1999, Dresden

Agenda item: **Ad hoc 1**

Source: **Siemens**

Title: **NodeB synchronisation for TDD**

Document for: **Discussion**

Introduction

At the last WG1#8 meeting in New York Tdoc R1-99g42 was presented, that introduced a method for nodeB synchronisation via special bursts in the RACH timeslot. Questions were raised about further technical details of the algorithm and the feasibility of such a method.

This paper tries to answer these questions and presents simulations results to demonstrate the performance of nodeB synchronisation via the RACH timeslot

Description of the method

The method to synchronise Node Bs together is based on using infrequent transmissions of sync bursts in the PRACH time slots. Typically, one such transmission would be made per second in a given region – Any given Node B would make one such transmission approximately about once every 20 sec. Such soundings between neighbour Node Bs facilitate timing offset measurements and allow frequency corrections. The timing offset measurements are reported back to the RNC for processing to generate Node B timing updates

The whole process is initiated and conducted by the RNC. It is assumed that, prior to over the air synchronisation, the RNC will have already synchronised the Node Bs to the accuracy available given the unpredictable network delays. This is taken to be ±50 ms although the actual figure is not critical. The RNC instructs each Node B when to transmit a sync burst in its PRACH slot and when to listen for a sync burst in a PRACH slot. Because of the uncertainty at both ends, the Node B must listen, initially performing correlations of the sync burst over a period of 200 ms, corresponding to 768,000 chip positions. The complexity of this process can be reduced by using a hierarchical code for the PRACH sync burst. Offline processing could be used to reduce real time processing requirements during the initial phase where the measurement windows are wide. No other Layer 1 processing is active during this initial phase.

A threshold is set for reception of the PRACH sync bursts at a Node B. If the received signal level does not exceed this threshold, no time measurement is reported.

Before the network is synchronised the Node Bs are only allowed to transmit their scheduled PRACH sync bursts. This is preferable to using SCH transmissions because it avoids the UE's synchronising prematurely to Node Bs which have not, themselves, become synchronised.

Simulation

A discrete event simulation was written to explore the operation of synchronisation.

Within the model, each Node B contains a clock. This can exhibit inaccuracies in all of three ways:

- A current time shift with respect to absolute time
- A rate error (frequency error) – This is intended to account for small unpredictable short to medium term effects
- A random error whose variance grows with time

The timing and the rate of the clocks can be adjusted at any time.

A deployment of Node Bs is established according to an approximate hexagonal grid. The radius of each hexagon is 1 km. Each Node B is fixed at the centre of a hexagon with x & y offsets drawn separately from a zero mean Gaussian distribution with a standard deviation of 50 m. Thus the average range to a nearest neighbour Node B is 2 km. The topological size of this deployment is specified in terms of the number of rings (tiers) of Node Bs surrounding a central Node B. The results are given for 5 tiers corresponding to 91 Node Bs. The connectivity is established according to a COST 231 Hata model and the following parameters:-

Parameter	Value
Carrier Frequency	2.0 GHz
Node B Antenna Height	5 m
Antenna Gain (Elevation)	6 dB
Environment Type	Urban
City Size	Large
Transmit Power	3 W
Noise Bandwidth	3.86 MHz
Processing Gain	2400
Receiver Noise Figure plus Receiver Feeder Loss	8 dB
Lognormal Standard Deviation	8 dB
Minimum Acceptable Despread Mean S/N	20 dB

According to the connectivity, the RNC (on the basis of assumed soundings) establishes a re-use pattern for sync transmissions. This is based on dividing the Node Bs into sets which have no common neighbours. For the deployment simulated, 23 such sets were determined. On this basis, every Node B makes a transmission once every 23 seconds. The number of such sets increases with transmit power. In the limit, where every Node B could hear every other Node B, the number of sets would be equal to the total number of Node Bs. All Node Bs in a set transmit sync bursts simultaneously. One second later, all Node Bs in the next set transmit simultaneously until all sets have transmitted, when the process then repeats from the first set.

Upon receipt of a sync burst, a Node B compares the timing of the burst to its own. Full account is taken of the propagation delays. All timing differences are reported back to the RNC. This accumulates the measurements into a matrix.

Periodically (every 10 seconds) the RNC performs the algorithm described in Appendix A to compute the updates to bring the Node Bs into synchronisation. The updates are then signalled to the relevant Node Bs.

When a Node B receives its update, it signals back to the RNC to confirm this so that the RNC can update its log of measurements accordingly. This is critical because a two-way sounding is required (because of propagation delay) to obtain the time difference between any two Node Bs. If a timing update takes place at one of the Node Bs after a sounding in one direction and before the sounding in the other, the difference will be inconsistent. Even if this does not happen, the actual time difference will have changed and if this is not compensated for and not updated by measurements before use for the next timing update, erroneous updates will result.

As described so far, the system can correct the timing errors but can do nothing about the clock ratio errors.

However, the rate and size of updates provides a measure of the clock rate error allowing the clocks at the Node B to be adjusted to take out the mean ratio error. This need not be done by adjusting the physical clock but could be achieved by inserting/deleting clock pulses at the required rate

If such clock rate adjustments are made too early in the sync procedure, inappropriate values will be performed. This is avoided by computing the adjustment and comparing it with the maximum specified error for the clock modules in use. If the computed adjustment exceeds this maximum then no adjustment is performed at all. As the system achieves tighter and tighter synchronisation, the computed adjustments will eventually fall within the maximum limits. When an adjustment fall within these limits, a fraction of it is applied to the clock as a correction.

Since all Node Bs are static, the path between them should also be non fading. However, moving reflectors in the path will still cause some occasional fading. For this reason the earliest path was modelled as Rician with k factor of 6 dB

Results are shown in Figure 1, below, for the following case

Parameter	Value
Initial Timing Error	Uniform Random Distribution over ± 50 ms
Initial Clock Rate Error	Uniform Random Distribution over ± 0.050 ppm
Measurement Resolution	¼ chip
Clock Variance	$10^{-17}sec^2/sec$
Timing Measurement Window	768000 chips about correct position
Clock Update Coefficient	20% of Measurement

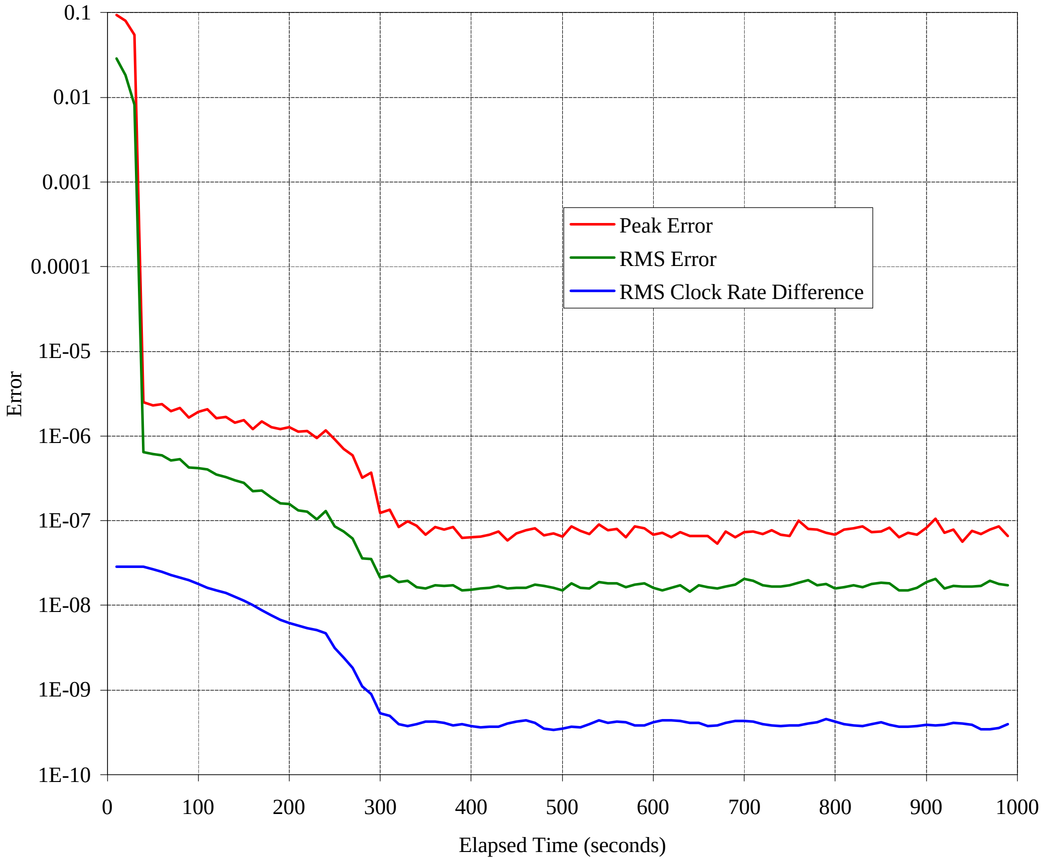

Figure 1 Sync Settling Performance

The initial sync period (the first 40 seconds) removes the initial error. However, with the initial clock rate errors, the update rate is too slow to compensate for the drift. Over the next 300 seconds the clock rates are gradually brought together so that, in the steady state, the final error is significantly reduced. This, in turn, reduces the timing errors between users. The peak error is defined as the largest error anywhere in the deployment between any Node B and any of its neighbours at the specified time. The RMS Clock Rate difference is the rms difference between all clocks. For the above case the worst case error is always close to 0.1 µs. This would be acceptable for location services.

More frequent PRACH sync bursts would bring the system into sync proportionally more rapidly. Because the system adjusts the clock rates in addition to the clock errors, the steady state error will be small. The exact minimum update rate will depend on the time instability of the clocks. An attempt has been made to model this by incorporating the random element. Less frequent updating would increase the peak error. For the current value of randomness parameter, more frequent updating would not significantly improve performance. Detailed measurements on representative clocks would be needed to establish the true value.

For the propagation model used the signal to noise ratio obtained is adequate. If the SCH were used it would need to be transmitted at lower power and for a shorter period (lower processing gain). This would make the reliability of detection significantly poorer, requiring validation over multiple frame reception.