A Distributed Synchronization Algorithm for Femtocells Network

Ligang Liu[1], Jiang Wang[1,2], and Jing Xu[1,2]

[1] Shanghai Research Center for Wireless Communication, Shanghai, 200335, China
[2] Key Laboratory of Wireless Sensor Network & Communication,
Shanghai Institute of Microsystem and Information Technology,
Chinese Academy of Sciences (CAS), Shanghai 200335, China
`{ligang.liu,jiang.wang,jing.xu}@shrcwc.org`

Abstract. A distributed algorithm is proposed to synchronize femtocells through air interface. A femtocell could listen to the synchronization signals of its neighbouring base stations to extract time information of these neighbours. Then it updates its time according to the neighbours' time, directed by a global optimization criterion in a distributed manner. Finally all femtocells will have same time information. A common framework is summarized to deal with many scenarios with different configuration of weight coefficients.

Keywords: Wireless communication, femtocell, synchronization, gossip algorithm, distributed communication.

1 Introduction

Synchronization among femtocells is very important [1, 2]. Just like a typical macrocell, femtocells require a certain level of synchronization (frequency or phase/time). On the one hand, frequency synchronization is required to ensure a tolerable carrier offset. On the other hand, time synchronization is of great importance especially for time division duplex (TDD) systems. It is necessary to align received signals, otherwise inter-slot interference will occur. Furthermore, synchronization is also required for macrocell users to handover to a femtocell, or vice versa [3–5].

Many solutions have been developed to synchronize base stations in wireless communication systems. GPS is the most popular synchronization method for current wireless networks because of its maturity and convenience. For a femtocell deployed inside buildings, a stable satellite signal indoor will be very weak so that it will take long time to obtain synchronization information, even lead to receivers out of work. Alternatively, femtocells could achieve synchronization through the backbone connection using IEEE 1588 PTP (Precision Time Protocol). However, PTP could suffer delay jitter because the delays on the Internet has a relative large dynamic range depending on the traffic, which is unpredictable. Another challenge for PTP is that it requires new investment to deploy

H. Qian and K. Kang (Eds.): WICON 2013, LNICST 121, pp. 96–103, 2013.

PTP-enabled routers throughout the path between the servers and the clients. In a word, it is difficult for PTP to be directly applied in femtocell.

An approach is proposed to employs the clock drift ratio information to achieve synchronization between non-interacting femtocells and macrocells [6]. But it requires information of user-equipments.

Another approach is to listen to the synchronization signals of neighbouring macrocells to synchronize the clock. This method is normally called *network listening* [7, 8]. For a femtocell, an efficient solution would be to listen to the nearest macrocell. Unfortunately, in a scenario where the macrocell coverage is poor, this method will be out of work. In the case of dense femtocells deployment, it is possible that the coverage of many cells will overlap each other. An acceptable solution for the synchronization could be to use the neighbouring femtocells, not only listening to the macrocells.

Alternatively, femtocells could achieve synchronization in a distributed manner. By listening to neighbouring femtocells, a femtocell calculates its synchronization time using a well-defined algorithm on certain criterion. This method is efficient especially when there are a large amount of nodes. These kinds of algorithms, so called gossip algorithms, have been studied in sensor networks to distribute the information among the different nodes [9].

In this paper we propose a synchronization scheme of femtocells in a distributed way in the case of dense femtocell deployment. The remainder of this article is organized as follows. In Section 2 we present a distributed synchronization algorithm for femtocells network along with analysis and discussion of some simple extensions. Section 3 presents computer simulations to evaluate performance of the proposed algorithm, and finally Section 4 concludes the paper.

2 Distributed Synchronization Algorithms for Femtocells

2.1 System Model

Let the femtocell network is composed of J femtocells, and we get a set of J nodes $V = \{v_1, v_2, \cdots, v_J\}$. There is a set of edges $E = \{e_{ij}\}$, where $e_{ij} \in E$ if and only if node v_i can listen to synchronization signal from node v_j. It leads to a graph $G = \{V, E\}$. Note that G is a directed graph because $e_{ij} \in E$ does not imply $e_{ji} \in E$. This would happen in the case where different nodes have different transmission power, or different link has different path loss. Every node v_i carries information t_i, which represents synchronization time of node v_i. It is assumed that the initial value of $t_i(0)$ is uniformly distributed in a limited range $[-c, c]$, where c is a positive number. The goal of the problem is to update t_i iteratively so that all nodes achieve same value as soon as possible, $t_1(n) = t_2(n) = \cdots = t_J(n)$ when n is greater than a certain number. Here n is the iteration number.

2.2 Distributed Synchronization Algorithm

At any iteration n, node v_i senses its neighbour node v_j and estimates time of v_j as t_{ij} where $1 \leq j \leq J (j \neq i)$. In order to determine new time, node v_i tries to optimize a cost function. An efficient cost function $u(t_i)$ could be the weighted sum of squared time difference between v_i and v_j,

$$u(t_i) = \sum_{j=1}^{J} \alpha_{ij} \left(t_i - t_{ij}\right)^2 . \tag{1}$$

Here α_{ij} is a weighting coefficient and note that for a node v_j that v_i cannot sense, $\alpha_{ij} = 0$. The existing of α_{ij} is because time information measured over different link should be assigned different weight. For example, time sensed from a strong synchronization signals should be assigned more weight than that from a weak neighbour, thus rendering the algorithm robust against measurement errors. It is necessary to pose a constraint, $\sum_{j=1}^{J} \alpha_{ij} = 1$.

A reasonable α_{ij} could be defined as ratio of total received synchronization signal power to that from node v_j,

$$\alpha_{ij} = \frac{P_{ij}}{\sum_{k=1}^{J} P_{ik}} \quad (i \neq j) \tag{2}$$

where P_{ij} is the synchronization signal power that node v_i has received from node v_j, and it is assumed here $\alpha_{ii} = 0$.

The distributed synchronization problem is to find an optimal t_i so that it can minimize $u(t_i)$ defined in equation (1) at every iteration for every node. It can be written as

$$t_{i,opt} = \arg\min_{t_i} u(t_i) \tag{3}$$

It is a simple problem and its solution is $t_{i,opt} = \sum\limits_{j=1}^{J} \alpha_{ij} t_{ij}$.

Ideally, t_{ij}, i.e. time of node v_j that node v_i has sensed, is exactly equal to t_j. However, estimation error z_{ij} is inevitable and $t_{ij} = t_j + z_{ij}$. Here we assume that z_{ij} is a zero-mean Gaussian variable with variance of σ_{ij}^2 and we assume that z_{ij} is independent to t_j. As a result, the time updating equation for node v_i from iteration $n-1$ to n is written as in a iterative way:

$$t_i(n) = \sum_{j=1}^{J} \alpha_{ij} [t_j(n-1) + z_{ij}]. \tag{4}$$

Note that here we have removed the effect of time component that has proceeded during an iteration, for the sake of concise. A femtocell could update its synchronization time in two ways, i.e. in synchronous way and asynchronous way.

2.3 Convergence Analysis of Synchronous Updating

In synchronous way, at every iteration, all of the femtocells update its time at same time. Here we temporarily assume that for all nodes, their weight coefficients are invariant.

For the sake of concise, we consider estimation error free algorithm first, i.e. $z_{ij} = 0$. Let $\mathbf{T}(n) = \{t_1(n), \ldots, t_J(n)\}^T$ and $\boldsymbol{\alpha}$ is a $J \times J$ matrix, $\boldsymbol{\alpha} = \{\alpha_{ij}\}$, the above equation is rewritten as

$$\mathbf{T}(n) = \boldsymbol{\alpha}\mathbf{T}(n-1) \tag{5}$$

Extend this series, it becomes $\mathbf{T}(n) = \boldsymbol{\alpha}^n\mathbf{T}(0)$. The convergence behavior of the algorithm is determined by properties of weight coefficient matrix $\boldsymbol{\alpha}$. It is obvious that $\boldsymbol{\alpha}$ is a row stochastic matrix, whose each row sum is unity. For a row stochastic matrix $\boldsymbol{\alpha}$, its largest eigenvalue is 1 and the others are absolutely smaller than 1. It is obvious that $\boldsymbol{\alpha}^n$ is convergent when $n \to \infty$ and $\boldsymbol{\alpha}^\infty$ is a $J \times J$ constant matrix $\mathbf{1}\mathbf{b}^T$, whose all rows are same to $\mathbf{b}^T$. Here $\mathbf{1}$ is a $J \times 1$ column vector that all elements are 1 and $\mathbf{b}$ is a $K \times 1$ column vector that is determined by eigenvector of α. Consequently, $\mathbf{T}(\infty) = \mathbf{b}^T\mathbf{T}(0)\mathbf{1}$. It means that the proposed algorithm is convergent and the final time value of all nodes is determined by $\boldsymbol{\alpha}$ and $\mathbf{T}(0)$.

The second order convergence of the proposed algorithm can be analyzed through $\xi = E\left\{||\mathbf{T}(n) - \mathbf{T}(\infty)||_2^2\right\}$. After some trivial mathematics manipulations, it can be drawn that $\xi = 0$.

Consider the case of non-zeroes estimation error, $z_{ij} > 0$, the same conclusion can be drawn after some trivial mathematics manipulations, remembering the independence assumption of z_{ij} and t_j. However, in this case, the second order convergence is different. It has $\xi = \sum_{i=1}^{J}\sum_{j=1}^{J}\alpha_{ij}^2\sigma_{ij}^2$. Because $0 \le \alpha_{ij} \le 1$, it can be derived that the synchronization error in distributed manner will be less than that of synchronization from only one neighbour. For example, if synchronization error of node v_1 from node v_2 or v_3 is $\sigma_{12}^2 = \sigma_{13}^2 = \sigma^2$, now let node v_1 gets synchronization time from both node v_1 and v_2 using above distributed algorithm, with same weight coefficients $\alpha_{12} = \alpha_{13} = 1/2$. Then the new synchronization error will be reduced to $\sigma^2/2$.

2.4 Convergence Analysis of Asynchronous Updating

In asynchronous way, at every iteration, only one femtocell updates its time and the others do nothing. It is assumed that at iteration n, node v_i synchronizes from node v_j. Let us construct a time-varying weight coefficients matrix $\alpha(n)$, which is a identity, except the i-th row is replaced by weight coefficients $\boldsymbol{\alpha}_i = \{\alpha_{i1}, \cdots, \alpha_{iJ}\}$. Note that $\boldsymbol{\alpha}(n)$ still is a row stochastic matrix. Hence the time updating equation is written as

$$\mathbf{T}(n) = \boldsymbol{\alpha}(n)\mathbf{T}(n-1) \tag{6}$$

Extend this series, it becomes $\mathbf{T}(n) = \prod_{m=n}^{1} \boldsymbol{\alpha}(m)\mathbf{T}(0)$.

Because $\boldsymbol{\alpha}(m)$ is row stochastic, it has been proved that the limit existed [9],

$$\lim_{n\to\infty}\prod_{m=n}^{1}\boldsymbol{\alpha}(m) = \mathbf{1}\mathbf{b}^T \tag{7}$$

It means that after certain n iterations $\prod_{m=n}^{1} \boldsymbol{\alpha}(m)$ has all its rows are identical. Hence $\mathbf{T}(n)$ has all its elements same as $\mathbf{b}^T\mathbf{T}(0)$. It should be noted that in asynchronous updating, the final value of $\mathbf{b}$ is determined by not only the value of $\boldsymbol{\alpha}(m)$ but also the order of $\boldsymbol{\alpha}(m)$.

Compared to the synchronous updating, the asynchronous updating will converge in a lower rate because it only synchronizes one node in one iteration while the synchronous method will update all nodes in one single iteration.

The proposed approaches are different from the gossip algorithms used in wireless sensor networks. In a gossip algorithm, in one iteration, a selected *pair* of neighboring nodes (v_i, v_j) exchange their current estimates and then update the estimates of both nodes as same value, $t_i(n) = t_j(n) = f(t_i(n-1), t_j(n-1))$, where f is a predefined function. The weight matrix in gossip algorithm usually is doubly stochastic. Although the doubly stochastic matrix has better characteristics than a row stochastic matrix, this condition is too strong for the synchronization problem. The the proposed algorithm in this paper is a kind of extension of the gossip algorithm and it is expected to be applied in more wide area.

2.5 Variant

The updating in equation (4) does not consider the current time of v_i itself. It would cause the time of v_i change dramatically. The time updating could use a moderate manner. For node v_i, another approach is to update its time based on its current time by adding a weighted item of time obtained from the neighbours. It can be written as

$$t_i(n) = \beta t_i(n-1) + (1-\beta)\Delta t_i(n-1) \tag{8}$$

$$\Delta t_i(n-1) = \sum_{j=1, j\neq i}^{J} \alpha_{ij}[t_j(n-1) + z_{ij}] \tag{9}$$

This modification can be represented accordingly by a new definition of $\boldsymbol{\alpha}$, where $\alpha_{ii} = \beta$ and

$$\alpha_{ij} = (1-\beta)\frac{P_{ij}}{\sum_{k=1}^{J} P_{ik}}, (j \neq i) \tag{10}$$

With this definition, it holds that $\sum_{j=1}^{J} \alpha_{ij} = 1$ for any $1 \leq i \leq J$, i.e. $\boldsymbol{\alpha}$ is still row stochastic. Consequently, this variant has similar convergence performance compared to the ones discussed in the previous subsections. The different is it will converge in a relatively lower rate, but in a smoother manner.

There should be a scenario where one node, v_i for example, obtain its synchronization from an external source, such as GPS. It is not necessary for it to synchronize from any neighbour, although it could provide synchronization to the other nodes. In this case, its weight coefficients should be defined as a zero vector except the i-th element is 1. The resultant $\boldsymbol{\alpha}$ is still row stochastic. It can be expected that in this case, when the algorithm has converged, all nodes has identical synchronization time same to that of v_i.

Summarily, the proposed algorithm can be represented by equation (6). Different definition of $\boldsymbol{\alpha}(m)$ will lead to different implementation of the proposed algorithm, which is suitable in the designed scenario.

3 Computer Simulations

In this section, performance of the proposed algorithm is verified by computer simulations. There are 20 femtocells randomly deployed in two building stripes. The transmission power of all femtocells are 20dBm.

$$PL(\mathrm{dB}) = 38.46 + 20\mathrm{log}_{10}D + 0.7d_{\mathrm{in}} + q_{\mathrm{i}}L_{\mathrm{iw}} + q_{\mathrm{o}}L_{\mathrm{ow}} \tag{11}$$

Here, D presents distance between two femtocells and d_{in} is indoor part (all distance is in meters). The item $0.7d_{\mathrm{in}}$ is energy loss caused by indoor materials. q_{i} is the number of inner walls between two femtocells and L_{iw} is energy loss through inner walls (5dB). q_{o} is the number of outdoor walls between two femtocells (0 for femtocells in same stripe and 2 for femtocells in different stripe) and L_{ow} is energy loss through outdoor walls (20dB).

3.1 Convergence and Influence of $\boldsymbol{\beta}$

Fig. 1 illustrates convergence in the first 30 iterations of the proposed synchronous updating algorithm with different parameter β. Each line represents the time of one femtocell during the convergence period. It can be seen that the algorithm can converge in very fast rate. The asynchronous updating method has similar performance, except it has much slower convergence rate, so the result is not presented here.

The parameter β is used to adjust behavior of the algorithm. Smaller the β is, less weight the current time poses to its new time. It can be seen from the two subfigures that when $\beta = 0.0$ the new time of each femtocell changes dramatically, while the convergence rate is faster than that of $\beta = 0.5$. Contrarily, the convergence of $\beta = 0.5$ looks smoother than that of $\beta = 0.0$.

In order to compare their convergence rate, the convergence performance is quantified using the definition as *deviation* $\chi(n) = \sum_{i=1}^{J}[t_i(n) - \bar{t}(n)]^2$, where $\bar{t}(n) = \sum_{i=1}^{J} t_i(n)/J$, i.e. the squared sum of difference between $t_i(n)$ and their average $\bar{t}(n)$.

Fig. 2 compares three $\chi(n)$ with $\beta = 0.0$, $\beta = 0.5$ and $\beta = 0.9$ respectively. It is obviously seen that smaller β has faster convergence rate.

3.2 Case of One Femtocell is Fixed

It has been discussed that when a femtocell has an external synchronization source, it is not necessary to update its time at any iteration but it can provide synchronization to the others. As a result, all the other femtocells can achieve synchronization of this special femtocell. Fig. 3 illustrates the convergence in this

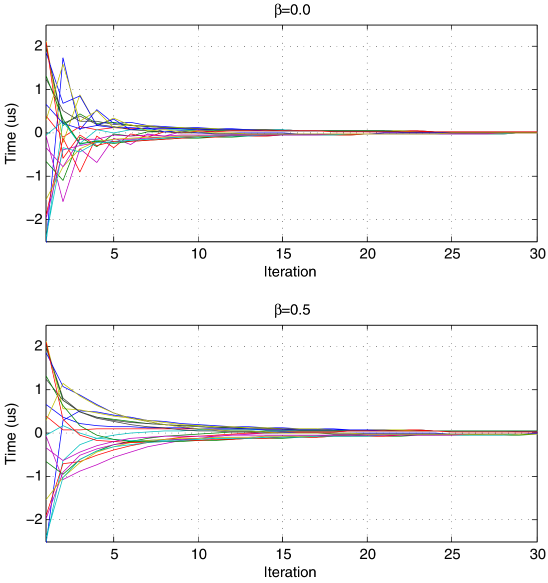

Fig. 1. Convergence of the proposed synchronous updating algorithm

case, where femtocell 1 located in the left bottom apartment has a fixed time 0. When the algorithm has converged, all femtocells have the same synchronization time of 0.

4 Conclusion

We have proposed a kind of distributed synchronization algorithms for femtocells network. Through minimizing the time difference between femtocells, the proposed algorithms can achieve all related femtocells be time synchronized. It is distinguished from the existing approaches that maintain synchronization only

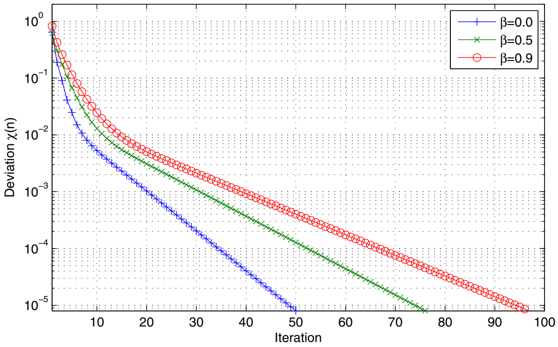

Fig. 2. Comparison of $\chi(n)$ with different β

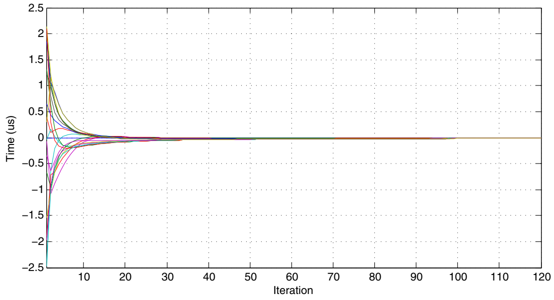

Fig. 3. Convergence in the case of one femtocell is fixed

with a single neighbour. The proposed algorithms get synchronization from multiple neighbours. This method can lead a cluster of femtocells synchronized in a faster rate than the method of single source. At the same time, this method has better synchronization accuracy than its single source counterpart. It does not require a central control node and it is self-organized and self-optimized.

Acknowledgments. This work was supported by National Science and Technology Major Project (NO. 2012ZX03003009-003), and project of Constructions of Innovative Service Platform for Wireless Mobile Communication Technology (No. 11DZ2290100), and project of New Generation Wireless Communication Network Architecture (No. 12JC1404202).

References

1. Zhang, J., Roche, G.: Femtocells: technologies and deployment. John Wiley & Sons, Ltd. (2010)
2. Chandrasekhar, V., Andrews, J.G.: Femtocell networks: a survey. IEEE Communications Magazine 46(9), 59–67 (2008)
3. 3GPP, Synchronization in UTRAN Stage 2 (Release 10), TS 25.402 (June 2011)
4. 3GPP, TDD Home eNode B Radio Frequency requirement analysis (Release 10), TR 36.922 (April 2011)
5. Jungnickel, V., Wirth, T., Schellmann, M., Haustein, T.: Synchronization of cooperative base stations. In: ISWCS (2008)
6. Mehrpouyan, H., Blostein, S.D., Svensson, T.: A New Distributed Approach for Achieving Clock Synchronization in Heterogeneous Networks. In: GLOBECOM, pp. 1–5 (2011)
7. Amin, P., Tirkkonen, O.: Network Listening based Synchronization techniques for Femtocell Systems. In: PIMRC, pp. 1–5 (2011)
8. Lien, S., Lee, H., Shih, S., Chen, S., Chen, K.: Network synchronization among femtocells. In: GLOBECOM, pp. 248–252 (2011)
9. Dimakis, A.G., Kar, S., Moura, J.M.F., Rabbat, M.G., Scaglione, A.: Gossip Algorithms for Distributed Signal Processing. Proceedings of the IEEE 98(11), 1847–1864 (2010)