Distributed Synchronization for OFDMA-Based Wireless Mesh Networks

Jihyung Kim, Jung-Hyun Kim, and Kwangjae Lim

In this paper, we propose a distributed synchronization algorithm for wireless mesh networks based on orthogonal frequency division multiple access. For time and frequency synchronization, a node requests its neighbor nodes for a change of fast Fourier transform starting points, transmission times, and carrier frequencies needed for synchronization. The node also updates its own time and frequency elements through simple formulas based on request messages received from neighbor nodes using a guard interval and a cyclic prefix. This process with the cooperation of neighbor nodes leads to a gradual synchronization of all nodes in the network. Through a performance comparison with a conventional scheme, we obtain simulation results indicating that the proposed scheme outperforms the conventional scheme in random topologies and a grid topology.

Keywords: Wireless mesh networks, distributed synchronization, cooperation, OFDMA.

Manuscript received Mar. 06, 2013; revised July 31, 2013; accepted Sept. 06, 2013.

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

Jihyung Kim (phone: +82 42 860 4828, savant21@etri.re.kr) and Kwangjae Lim (kjlim@etri.re.kr) are with the Communications & Internet Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Jung-Hyun Kim (jh.kim06@yonsei.ac.kr) is with the Department of Electronic and Electrical Engineering, Yonsei University, Seoul, Rep. of Korea.

I. Introduction

The recent demand for a wireless mobile backhaul network has increased the amount of research into wireless mesh networks (WMNs) with a long coverage [1], [2] owing to a reduction in cost and time, as well as infrastructure independence [3]. WMNs with a short coverage or centralized topology have been studied for variable standards, such as wireless local area networks [4], wireless metropolitan area networks [5], wireless personal area networks [6], and wireless sensor networks (WSNs) [7], helping to substantially improve the network performance, cut down the operation costs, and bring more convenience to both operators and end users. These standards are mainly focused on orthogonal frequency division multiplexing (OFDM) and based on carrier sense multiple access with collision avoidance or centralized scheduling and are therefore unsuitable for long-coverage and mobile networks [8].

On the other hand, WMNs using orthogonal frequency division multiple access (OFDMA) for an efficient resource allocation and long coverage have recently been considered [9]. OFDMA is a suitable multiplexing scheme for efficient network overhead and low latency [10]. Taking a long coverage into account, synchronization is one of the critical issues [11], [12], whereas, to the best of our knowledge, there have been few studies for OFDMA-based WMNs. Time synchronization algorithms for wireless ad hoc or sensor networks are generally focused on a synchronization of the transmission time (T) [13], [14]. This is because they do not consider the propagation delay from a short coverage or a simultaneous reception using OFDMA. The main concern for OFDMA-based WMNs is the synchronization of the receiving times. At an OFDMA receiver (RX), if the transmitter (TX) nodes are not synchronized with the RX node when multiple signals from nodes are transmitted simultaneously, they will interfere with each other because of the different propagation delays, and the RX node will not be able to recover the individual signal of each node [11]. Hence, for the OFDMA physical layer mode to work properly, all transmission signals should arrive at the RX node at the same time with high frequency accuracy. This can be achieved if all TX nodes are synchronized with the RX node before the communication link is established. In conventional cellular networks, synchronization with a preamble and ranging code based on the base station (BS) is adopted [15], [16]. However, this problem is more complex in WMNs because there are multiple pairs between TX nodes and RX nodes. Relevant research has been conducted on asynchronous OFDMA systems, but it mainly focused on multipoint-to-point communication scenarios [17]-[20].

In this paper, we propose a simple solution considering one-hop neighbor nodes (node n denotes a neighbor node) to solve this problem for OFDMA-based WMNs. Each node updates its fast Fourier transform (FFT) starting point (*FFT*) and *T* using simple formulas based on the receiving time (*t*) synchronization information with the guard interval (*G*, distinguished as left [$_L$] or right [$_R$]) between frames/subframes (frs/sfrs) and the cyclic prefix (CP). To reduce the gap of each *t*, which is different owing to the propagation delay from the distance between nodes, *G* and CP are applied. In addition, for a carrier frequency misalignment between multiple pairs of nodes, each RX estimates the carrier frequencies of one-hop neighbor nodes and controls its carrier frequency to the average value of estimates by sharing messages with neighboring nodes. This leads to the gradual global synchronization of all nodes in the network.

The rest of this paper is organized as follows. Section II gives a brief introduction of OFDMA-based distributed synchronization. Section III describes the ranging process and the proposed algorithm for the distributed synchronization in detail. Section IV gives the performance analysis for the proposed scheme. The convergence speed of the proposed scheme is also discussed. Finally, section V offers some concluding remarks regarding our proposal.

II. Synchronization in OFDMA-Based WMNs

In distributed networks, such as OFDMA-based WMNs, it is difficult to make all received signals align because all node pairs have different distances, different transmission times, and different carrier frequencies. The modification for one specific node pair affects all adjacent node pairs of the specific node pair [21]. In Fig. 1, for example, nodes a and c are not one-hop

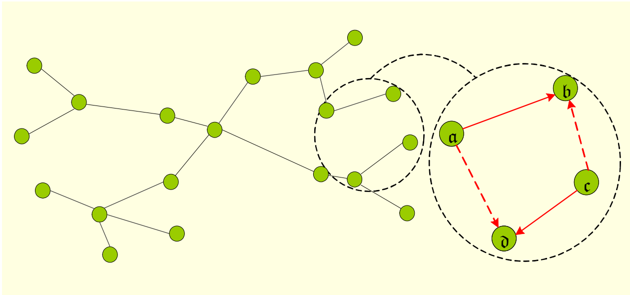

Fig. 1. WMN for simultaneous transmission of nodes a and c. Solid red line represents transmission of signal, and dashed red line represents interference at nodes b and d.

neighbor nodes, and scheduling and routing protocols thus allow simultaneous transmission. Assume that node a transmits a signal to node b and node c transmits a signal to node d at the same time, but each node is allocated with a different part of the bandwidth. If nodes a and c are not synchronized with each other in the time/frequency domain, the signal from node a and the signal from node c can interfere with node d and node b, respectively, although they have a dedicated bandwidth for each node through scheduling. This multiple pair synchronization problem is more complex than that of single pair synchronization and needs a dynamic solution according to the network topology.

To avoid interference in the time domain, all transmitted signals should arrive within the range of the effective CP (CP_E). Here, CP_E is the rest of the length of CP (*CP*), exclusive of the length of the channel delay spread (CP_M). In the frequency domain, there is no guard band for avoiding interference. If only one adjacent node has a misaligned carrier frequency, inter-carrier interference occurs because it cannot guarantee the orthogonality between subcarriers. For this, a global synchronization for at least one-hop neighbor nodes is needed.

The interference from the asynchronous signals of neighbor nodes is hard to handle through only scheduling and routing. In the next section, we present a novel synchronization method for solving this problem.

III. Distributed Synchronization

In this section, we propose a novel distributed synchronization method. The distributed synchronization method consists of distributed time synchronization and distributed frequency synchronization. We first describe a network entry mechanism for an estimation of the round trip delay (*RTD*) between nodes and the reference starting time (T_{rf}) for a fr/sfr in our considered scenario and then present the distributed time and frequency synchronization schemes in

http://dx.doi.org/10.4218/etrij.14.0113/0223

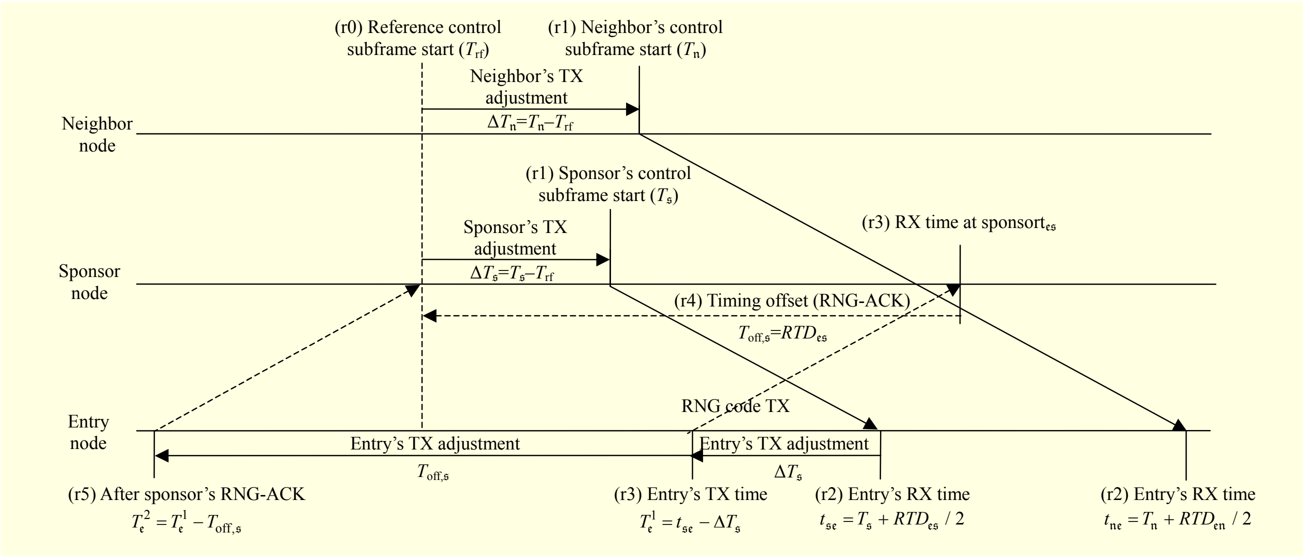

Fig. 2. Network entry and ranging process for estimation of RTD and T_{rf}.

detail.

1. Network Entry Mechanism

When a node enters a network, the node executes the ranging process for network entry similar to that of typical cellular systems [15], [16]. Once the entry node (node $\mathfrak{e}$) senses a sponsor node (node $\mathfrak{s}$) among neighbor nodes for network entry, it first scans for preambles and synchronizes itself with node $\mathfrak{s}$. Here, node $\mathfrak{n}$ is defined as a node that is exactly one-hop count away from the node of interest, and node $\mathfrak{s}$ is defined as the neighbor node that relays the data transmission to and from the interest node, which is closer to the BS or the backbone than the node itself. After synchronizing with node $\mathfrak{s}$ initially, node $\mathfrak{e}$ randomly selects a ranging code and then transmits the ranging code to node $\mathfrak{s}$. Node $\mathfrak{s}$ determines the time and frequency offset (F_{off}) of node $\mathfrak{e}$ from the received ranging code and transmits a ranging acknowledgement (RNG-ACK) message to node $\mathfrak{e}$.

We can examine the ranging in greater detail through the timing diagram shown in Fig. 2. Assuming an established network, node $\mathfrak{s}$ and node $\mathfrak{n}$ periodically broadcast $\Delta T_{\mathfrak{s}}$ and $\Delta T_{\mathfrak{n}}$ at (r1) $T_{\mathfrak{s}} = T_{rf} + \Delta T_{\mathfrak{s}}$ and (r1) $T_{\mathfrak{n}} = T_{rf} + \Delta T_{\mathfrak{n}}$, separately. $\Delta T_{\mathfrak{s}}$ and $\Delta T_{\mathfrak{n}}$ represent the T of node $\mathfrak{s}$ and node $\mathfrak{n}$ compared with T_{rf}, respectively, and they are determined on each update time by the distributed time synchronization scheme described in section III. If node $\mathfrak{e}$ receives preambles as well as broadcast messages with $\Delta T_{\mathfrak{s}}$ of node $\mathfrak{s}$ at (r2) $t_{\mathfrak{se}}$, node $\mathfrak{e}$ then sets the (r3) $T_{\mathfrak{e}}^1$ as a temporary starting point of a fr/sfr and afterwards transmits the ranging code to node $\mathfrak{s}$ at (r3) $T_{\mathfrak{e}}^1$. Node $\mathfrak{e}$ receiving the RNG-ACK message with (r4) $T_{off,\mathfrak{s}}$ can calculate RTD between node $\mathfrak{e}$ and node $\mathfrak{s}$ ($RTD_{\mathfrak{es}}$) and T_{rf}, as follows:

$$RTD_{\mathfrak{es}} = T_{off,\mathfrak{s}}, \qquad T_{rf} = T_{\mathfrak{e}}^1 - \frac{RTD_{\mathfrak{es}}}{2}. \tag{1}$$

Note that T_{rf} can be obtained from the Global Positioning System (GPS) [22], and (1) is applied when GPS is unavailable. In addition, node $\mathfrak{e}$ can calculate RTD between node $\mathfrak{e}$ and node $\mathfrak{n}$ ($RTD_{\mathfrak{en}}$) as follows:

$$RTD_{\mathfrak{en}} = 2 \cdot \{(t_{\mathfrak{ne}} - \Delta T_{\mathfrak{n}}) - (t_{\mathfrak{se}} - \Delta T_{\mathfrak{s}})\} + RTD_{\mathfrak{es}}. \tag{2}$$

Finally, node $\mathfrak{e}$ synchronizes its own time to node $\mathfrak{s}$ and sends a network entry request message including RTD information, $RTD_{\mathfrak{es}}$ and $RTD_{\mathfrak{en}}$, to node $\mathfrak{s}$ at (r5) $T_{\mathfrak{e}}^2$. Node $\mathfrak{s}$ can relay $RTD_{\mathfrak{en}}$ to node $\mathfrak{n}$. In addition, node $\mathfrak{e}$ can directly transmit $RTD_{\mathfrak{en}}$ to node $\mathfrak{n}$ at $T_{\mathfrak{e}}^3 = T_{\mathfrak{e}}^2 - \{(t_{\mathfrak{ne}} - \Delta T_{\mathfrak{n}}) - (t_{\mathfrak{se}} - \Delta T_{\mathfrak{s}})\}$.

2. Distributed Time Synchronization

Even though every node is synchronized with its sponsor node, interference can occur because of asynchronous signals from adjacent nodes. In other words, the fixed T and fixed t synchronized for one node cannot combat interference because the t by the distance for another node is different. Thus, a global synchronizing process for considering one-hop neighbor nodes is needed in distributed networks.

The simplest method for the avoidance of interference is using a long CP (CP_{long}), which covers the propagation delay by the distance between each node. Because CP is inserted in every OFDM symbol, however, this method requires a large overhead.

In this section, we propose distributed time synchronization, which is a solution for the above-mentioned problem. Distributed time synchronization controls the T and t per fr/sfr,

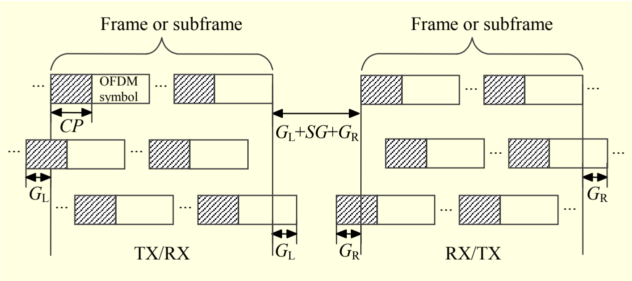

Fig. 3. Structure of fr/sfr with G_L and G_R for distributed time synchronization, including SG for flipping between reception and transmission.

not per symbol. Distributed time synchronization is done in two stages: the reception setting procedure and the transmission setting procedure. For the reception setting, each node sets its FFT based on the arrival times of the neighbor node signals. Similarly, for the transmission setting, each node sets T based on the reception times of the neighbor nodes. In the distributed time synchronization, to combat the interference, the FFT and T are dynamically changed by the request messages of the neighbor nodes. For this, the guard interval for changing the FFT/T is needed between frs/sfrs. Figure 3 shows this guard interval (G_L and G_R) between frs/sfrs, which is used for the early or late T/FFT and contains an effect of the propagation delay similar to CP_{long}. Here, SG is the latency time for flipping the reception and transmission. This has a lower overhead than using CP_{long} per symbol because a fr/sfr includes several OFDM symbols. Hereafter, we present the distributed time synchronization procedure in detail.

A. Reception setting Procedure

If an arbitrary node (node 𝔢) in the network performs the reception setting, its FFT range is then

$$FFT_L \le FFT_{\mathfrak{e}} \le FFT_R , \tag{3}$$

where $FFT_{\mathfrak{e}}$ is the FFT of node 𝔢, and FFT_L and FFT_R denote the minimum and maximum value of the possible FFT to receive, respectively. The FFT range is initially set to $FFT_L = T_{rf} - G_R$ and $FFT_R = T_{rf} + G_L + CP$.

To detect received signals from neighbor nodes with no interference, an intersection region should exist between the effective FFT range for the earliest received signal and the effective FFT range for the latest received signal. Here, the effective FFT range means the range from (t+CP_M) to (t+CP). This condition is shown in Fig. 4(a) and can be written as follows:

$$t_{i\mathfrak{e}} + CP_M \le FFT_{\mathfrak{e}} \le t_{j\mathfrak{e}} + CP , \tag{4}$$

where $i = \arg\max_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$ and $j = \arg\min_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$. In addition,

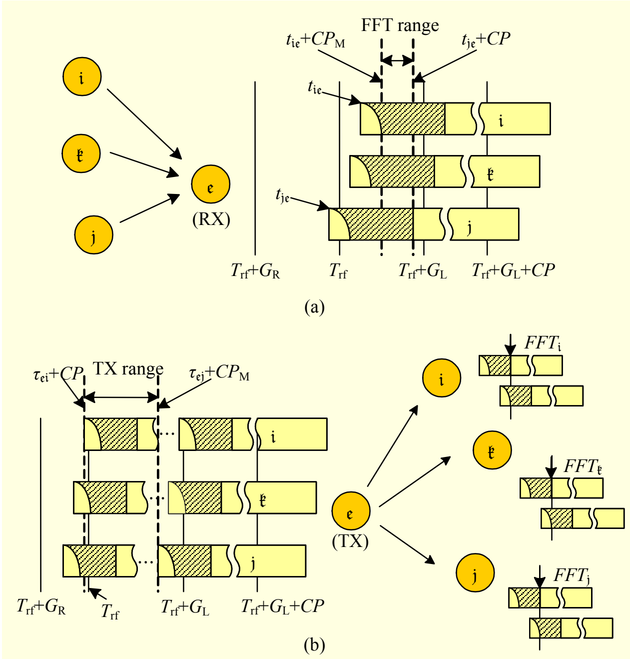

Fig. 4. Reception and transmission setting procedures in distributed time synchronization: (a) intersection region for reception setting; (b) intersection region for transmission setting.

$t_{\mathfrak{n}\mathfrak{e}}$ is the t for the signal of node 𝔫 at node 𝔢, $\mathfrak{n} \in \chi$, and χ is the alphabet set of one-hop neighbor nodes from node 𝔢.

Node 𝔢 excludes a specified node from the set of neighbor nodes by a selection rule in the case that (4) is not satisfied for all nodes of χ, and the intersection region is then recalculated. Two types in the exclusionary rule can be applied. First, a node of $\arg\max_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$ or $\arg\min_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$ is selected. Second, a node of $\arg\max_{\mathfrak{n}} \left(\left| \tilde{t}_{\mathfrak{e}} - t_{\mathfrak{n}\mathfrak{e}} \right| \right)$ is selected, where $\tilde{t}_{\mathfrak{e}}$ is the average value of all $t_{\mathfrak{n}\mathfrak{e}}$. For both types, node 𝔢 requests that the node of $\arg\max_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$ changes $T_{c\mathfrak{n}}$ into $T_{c\mathfrak{n}}+\Delta T_{cr}$ and the node of $\arg\min_{\mathfrak{n}} (t_{\mathfrak{n}\mathfrak{e}})$ changes $T_{c\mathfrak{n}}$ into $T_{c\mathfrak{n}}-\Delta T_{cr}$. This reduces the gap between the receiving times of the neighbor nodes at node 𝔢. Here, $T_{c\mathfrak{n}}$ denotes the reference value for the determination of the T of node 𝔫, and ΔT_{cr} is a request value for the T, of which the unit can be one or more samples in the time domain. In other words, the T of node 𝔫, which is described in the next part for the transmission setting procedure, is determined as the nearest value of $T_{c\mathfrak{n}}$ in the transmission range.

By (4), the updated FFT range is

$$\begin{aligned} FFT_L &= \max\left(T_{rf} - G_R,\ t_{i\mathfrak{e}} + CP_M\right), \\ FFT_R &= \min\left(T_{rf} + G_L + CP,\ t_{j\mathfrak{e}} + CP\right), \end{aligned} \tag{5}$$

and node 𝔢 sets $FFT_{\mathfrak{e}}$ as the nearest value of $FFT_{cr\mathfrak{e}}$ in the above range as follows:

http://dx.doi.org/10.4218/etrij.14.0113/0223

$$FFT_{\mathfrak{e}} = \begin{cases} FFT_{\mathrm{L}} & \text{if } FFT_{\mathrm{cre}} < FFT_{\mathrm{L}}, \\ FFT_{\mathrm{R}} & \text{else if } FFT_{\mathrm{cre}} > FFT_{\mathrm{R}}, \\ FFT_{\mathrm{cre}} & \text{otherwise,} \end{cases} \quad (6)$$

where FFT_{cre} indicates the reference value for the determination of the $FFT_{\mathfrak{e}}$, its initial value can be set to $T_{\mathrm{rf}}+CP$, and it is updated by the request of neighbor nodes through the transmission setting procedure as follows:

$$FFT_{\mathrm{cre}}^{u} = FFT_{\mathrm{cre}}^{(u-1)} + M_{\mathrm{F}} \times \Delta FFT_{\mathrm{cr}}, \quad (7)$$

where $FFT_{\mathrm{cre}}{}^{u}$ denotes FFT_{cre} of the u-th update time and ΔFFT_{cr} is a request value for the FFT, of which the unit can be one or more samples in the time domain. M_{F} is +1 if the number of request messages for ΔFFT_{cr} is more than that of $-\Delta FFT_{\mathrm{cr}}$, and M_{F} is –1 if vice versa. FFT_{cre} is limited to the initial values of FFT_{L} and FFT_{R}. That is,

$$FFT_{\mathrm{cre}}^{u} = \begin{cases} T_{\mathrm{rf}} - G_{\mathrm{R}} & \text{if } FFT_{\mathrm{cre}}^{u} < T_{\mathrm{rf}} - G_{\mathrm{R}}, \\ T_{\mathrm{rf}} + G_{\mathrm{L}} + CP & \text{if } FFT_{\mathrm{cre}}^{u} > T_{\mathrm{rf}} + G_{\mathrm{L}} + CP. \end{cases} \quad (8)$$

The exemplary algorithm of the reception setting procedure on the u-th update time in the pseudocode is stated in Algorithm 1, in which the excluded node by the exclusionary rule is $\arg\max_{\mathfrak{n}}(t_{\mathfrak{n}\mathfrak{e}})$. The number of one-hop neighbor nodes of node $\mathfrak{e}$ is denoted by $O_{\mathfrak{e}}$, which is also the total number of elements of χ.

Algorithm 1. Reception setting for distributed time synchronization.

```
Initialize FFT_L, FFT_R, FFT_cre
  χ̂ = χ, RX_setting_flag = 0, count = 1
  while RX_setting_flag = 0, O_e != 0
    for every n ∈ χ̂ do
      Find i, j where i = arg max_n (t_ne), j = arg max_n (t_ne)
    end for
    if t_ie+CP_M ≤ t_je+CP, t_ie+CP_M ≤ FFT_R, FFT_L ≤ t_je+CP
      Set RX_setting_flag = 1
    else
      if count = 1
        Set T_crj = T_crj + ΔT_cr
      end if
      Set T_cri = T_cri – ΔT_cr
      χ̂ = χ̂ \ i, O_e = O_e – 1, count = count + 1
    end if
  end while
  if O_e = 0
    Set FFT_L = T_rf – G_R, FFT_R = T_rf + G_L + CP, FFT_e = FFT_cre
  else
 Set FFT_L = max(T_rf – G_R, t_ie + CP_M),
      FFT_R = min(T_rf + G_L + CP, t_je + CP),
      FFT_cre^u = FFT_cre^(u–1) + M_F × ΔFFT_cr
    if FFT_cre^u < FFT_L
      Set FFT_e = FFT_L
    else if FFT_cre^u > FFT_R
      Set FFT_e = FFT_R
    else
      Set FFT_e = FFT_cre^u
    end if
  end if
  if FFT_cre^u < T_rf – G_R
    Set FFT_cre^u = T_rf – G_R
  end if
  if FFT_cre^u > T_rf + G_L + CP
    Set FFT_cre^u = T_rf + G_L + CP
  end if
```

B. Transmission Setting Procedure

The transmission setting performs a similar approach as the reception setting procedure. If node $\mathfrak{e}$ performs the transmission setting, its T range is then

$$T_{\mathrm{L}} \le T_{\mathfrak{e}} \le T_{\mathrm{R}}, \quad (9)$$

where $T_{\mathfrak{e}}$ is the T of node $\mathfrak{e}$, and T_{L} and T_{R} are the minimum and maximum values of the T range, respectively. The T range is initially set to $T_{\mathrm{L}} = T_{\mathrm{rf}} - G_{\mathrm{R}}$ and $T_{\mathrm{R}} = T_{\mathrm{rf}} + G_{\mathrm{L}}$.

As in the preceding, an intersection region should exist between the effective transmission ranges based on the FFT of each node $\mathfrak{n}$ such that each node $\mathfrak{n}$ receives the signal on its FFT. The effective transmission range means the range from(virtual transmission value – CP) to (virtual transmission value $-$ CP_{M}), where the virtual transmission value is the propagation delay subtracted from the FFT of node $\mathfrak{n}$. This condition is shown in Fig. 4(b) and can be expressed as follows:

$$\tau_{\mathfrak{e}i} - CP \le T_{\mathfrak{e}} \le \tau_{\mathfrak{e}j} - CP_{\mathrm{M}}, \quad (10)$$

where $\tau_{\mathfrak{e}\mathfrak{n}} = FFT_{\mathfrak{n}} - RTD_{\mathfrak{e}\mathfrak{n}}/2$, $i = \arg\max_{\mathfrak{n}}(\tau_{\mathfrak{n}\mathfrak{e}})$, $j = \arg\min_{\mathfrak{n}}(\tau_{\mathfrak{e}\mathfrak{n}})$, and $n \in \chi$.

When (10) is not satisfied for χ, node $\mathfrak{e}$ excludes a node from χ according to the exclusionary rule using the same approach with the reception setting procedure. In addition, node $\mathfrak{e}$ requests that the node of $\arg\min_{\mathfrak{n}}(\tau_{\mathfrak{e}\mathfrak{n}})$ changes FFT_{cm} into $FFT_{\mathrm{cm}} + \Delta FFT_{\mathrm{cr}}$ and the node of $\arg\max_{\mathfrak{n}}(\tau_{\mathfrak{e}\mathfrak{n}})$ changes FFT_{cm} into $FFT_{\mathrm{cm}} - \Delta FFT_{\mathrm{cr}}$. This process will be continued until (10) is satisfied. The updated T range with (10) is

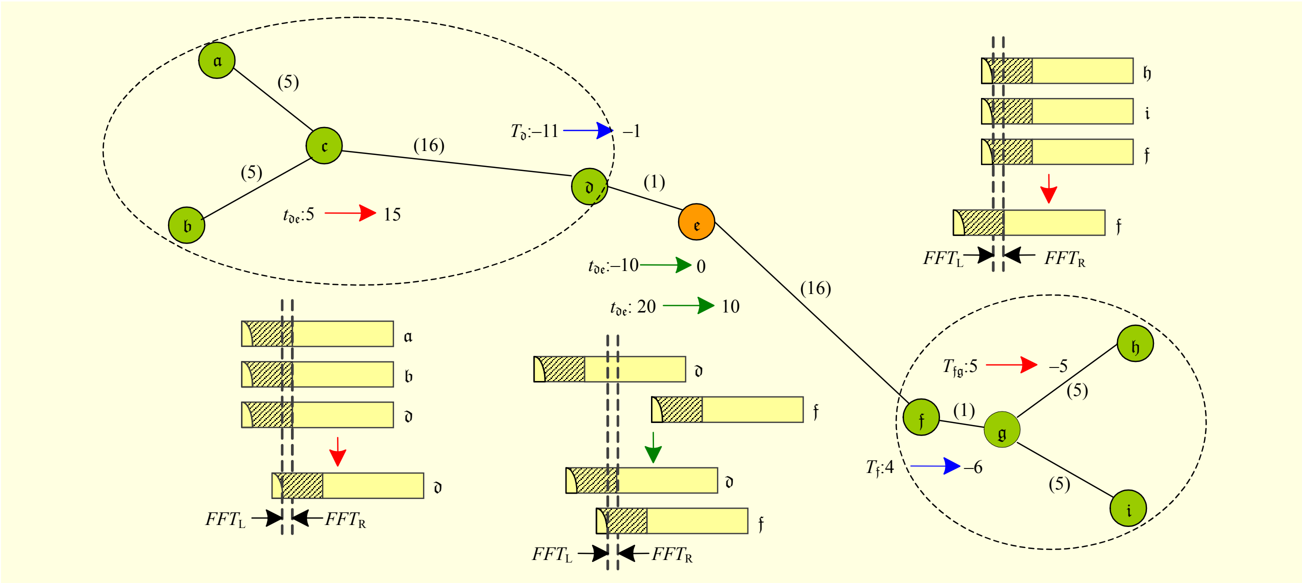

Fig. 5. Example for the distributed time synchronization: the change of nodes $\mathfrak{d}$ and $\mathfrak{f}$ due to node $\mathfrak{e}$, (·): the propagation delay(μs) between nodes.

$$T_L = \max\left(T_{rf} - G_R,\ \tau_{ei} - CP\right),$$
$$T_R = \min\left(T_{rf} + G_L,\ \tau_{ej} - CP_M\right), \quad (11)$$

and T_{cre} at the u-th update time, of which the initial value can be set to T_{rf}, is determined by the request of neighbor nodes through the reception setting procedure as follows:

$$T_{cre}^{u} = T_{cre}^{(u-1)} + M_T \times \Delta T_{cr}, \quad (12)$$

where M_T is +1 if the number of request messages for ΔT_{cr} is more than that of $-\Delta T_{cr}$, and M_T is −1 if vice versa. Node $\mathfrak{e}$ then sets $T_{\mathfrak{e}}$ as the nearest value of T_{cre} in the above range as follows:

$$T_{\mathfrak{e}} = \begin{cases} T_L & \text{if } T_{cre} < T_L, \\ T_R & \text{else if } T_{cre} > T_R, \\ T_{cre} & \text{otherwise,} \end{cases} \quad (13)$$

and T_{cre} is also limited to the initial values of T_L and T_R because the affordable gap between frs/sfrs is G_L+G_R.

$$T_{cre}^{u} = \begin{cases} T_{rf} - G_R & \text{if } T_{cre}^{u} < T_{rf} - G_R, \\ T_{rf} + G_L & \text{if } T_{cre}^{u} > T_{rf} + G_L. \end{cases} \quad (14)$$

The exemplary algorithm of the transmission setting procedure on the u-th update time in the pseudocode when the excluded node by the exclusionary rule is $\arg\min_{n}\left(\tau_{en}\right)$ is stated in Algorithm 2.

Figure 5 is a simple example of the distributed time synchronization. In this example, CP_E is 11 µs. Assume a scenario in which node $\mathfrak{e}$ enters between the two established networks. In the first network, nodes $\mathfrak{a}$ and $\mathfrak{b}$ transmit signals at 0 µs, node $\mathfrak{d}$ transmits a signal at −11 µs, and node $\mathfrak{c}$ receives

Algorithm 2. Transmission setting for distributed time synchronization.

Initialize T_L, T_R, T_{cre}
 $\hat{\chi} = \chi$, *TX_setting_flag* = 0, *count* =1
while *TX_setting_flag* = 0, $O_{\mathfrak{e}}$!=0
 for every $n \in \hat{\chi}$ **do**
  Find i, j where $i = \arg\max_{n}\left(\tau_{en}\right)$, $j = \arg\max_{n}\left(\tau_{en}\right)$
 end **for**
 if $\tau_{ei} - CP \le \tau_{ej} - CP_M$, $\tau_{ei} - CP \le T_R$, $T_L \le \tau_{ej} - CP_M$
  Set *TX_setting_flag* =1
 else
  if *count* =1
   Set FFT_{cri}=FFT_{cri}−ΔFFT_{cr}
  end **if**
  Set FFT_{crj}=FFT_{crj}+ΔFFT_{cr}
  $\hat{\chi} = \hat{\chi} \setminus j$, $O_{\mathfrak{e}}$=$O_{\mathfrak{e}}$−1, *count* =*count*+1
 end **if**
end **while**
if $O_{\mathfrak{e}}$= 0
 Set T_L=T_{rf}− G_R, T_R=T_{rf}+G_L, $T_{\mathfrak{e}}$=T_{cre}
else
 Set T_L=max(T_{rf}− G_R, $\tau_{ei} - CP$),
  T_R=min(T_{rf}+G_L, $\tau_{ej} - CP_M$
  T_{cre}^{u}=$T_{cre}^{(u-1)}$ +M_T×ΔT_{cr}
 if T_{cre}^{u}<T_L
  Set $T_{\mathfrak{e}}$=T_L
 else if T_{cre}^{u}>T_R

http://dx.doi.org/10.4218/etrij.14.0113/0223

Set $T_{\mathfrak{e}}=T_{\mathrm{R}}$
else
Set $T_{\mathfrak{e}}=T_{\mathrm{cre}}^{u}$
end **if**
end **if**
if $T_{\mathrm{cre}}^{u}<T_{\mathrm{rf}}-G_{\mathrm{R}}$
Set $T_{\mathrm{cre}}^{u}=T_{\mathrm{rf}}-G_{\mathrm{R}}$
end **if**
if $T_{\mathrm{cre}}^{u}>T_{\mathrm{rf}}+G_{\mathrm{L}}$
Set $T_{\mathrm{cre}}^{u}=T_{\mathrm{rf}}+G_{\mathrm{L}}$
end **if**

all the signals at 5 μs because of the propagation delay of the link between nodes $\mathfrak{a}$, $\mathfrak{b}$, and $\mathfrak{d}$. The propagation delay is denoted by (·) in the figure. In the second network, nodes $\mathfrak{h}$ and $\mathfrak{i}$ transmit signals at 0 μs, node $\mathfrak{f}$ transmits a signal at 4 μs, and node $\mathfrak{g}$ then receives all signals at 5 μs. If node $\mathfrak{e}$ enters between two networks, it receives the signal from node $\mathfrak{d}$ at –10 μs and receives the signal from node $\mathfrak{f}$ at 20 μs. Node $\mathfrak{e}$ cannot receive both signals because the gap between the arrival times of the two signals is over CP_{E}. In this situation, node $\mathfrak{e}$ requests changing T of node $\mathfrak{d}$ and that of node $\mathfrak{f}$. They can then change T in their admission range. If nodes $\mathfrak{d}$ and $\mathfrak{f}$ gradually change T to –1 μs and –6 μs, respectively, node $\mathfrak{e}$ can then receive both signals by forming the intersection range for the *FFT*. In addition, existing links for nodes $\mathfrak{c}$ and $\mathfrak{g}$ are maintained because the gap between arriving times is within CP_{E}. The update process of the *FFT* is performed side by side, though the description is omitted.

3. Distributed Frequency Synchronization

As mentioned in section III, node $\mathfrak{e}$ synchronizes with node $\mathfrak{s}$ through the ranging process. For the ranging process, node $\mathfrak{e}$ receives preamble signals of node $\mathfrak{s}$ several times and transmits a ranging code synchronized by preamble signals. Node $\mathfrak{s}$ estimates F_{off} and sends an RNG-ACK message, including the F_{off} information, to node $\mathfrak{e}$. Node $\mathfrak{e}$ controls its carrier frequency using F_{off} obtained from the feedback message of node $\mathfrak{s}$.

On the other hand, if the node controls F_{off} according to node $\mathfrak{s}$, F_{off} values of the neighbor nodes exclusive of node $\mathfrak{s}$ can then arise owing to an imperfect offset estimation, oscillator error, velocity, and so on. It is therefore necessary to synchronize in a global manner. For this, each node periodically performs the distributed frequency synchronization process after network entry [23].

The process offers a gradual synchronization through two steps: averaging through repetitive estimation and sharing the estimates. Node $\mathfrak{e}$ receives shared messages of the neighbor nodes, and repetitively estimates F_{off} values of neighbor nodes using a preamble. Node $\mathfrak{e}$ calculates the average value using both shared messages and repetitive estimated F_{off} values and then controls its own carrier frequency to the average value. As the next step, node $\mathfrak{e}$ sends shared messages, which are the differences between the average value and estimated offsets from the neighbors. This value is used for neighbor nodes in the next update.

The distributed frequency synchronization process on the u-th update time is presented in Algorithm 3. In the algorithm, $R_{\mathfrak{n}}$ is the number of preamble receptions of node $\mathfrak{n}$'s between the period of (u–1) and u, $\tilde{f}_{\mathfrak{en}}^{u}(r)$ is the estimated F_{off} between node $\mathfrak{e}$ and node $\mathfrak{n}$ at the r-th preamble reception, and $\tilde{f}_{\mathfrak{en}}^{u}$ is the average F_{off} of $\tilde{f}_{\mathfrak{en}}^{u}(r)$ after the $R_{\mathfrak{n}}$ reception. If node $\mathfrak{e}$ receives $s_{\mathfrak{ne}}^{(u-1)}$ before the u-th update, it calculates $\hat{f}_{\mathfrak{en}}^{u}$ with $\tilde{f}_{\mathfrak{en}}^{u}$ and $s_{\mathfrak{ne}}^{(u-1)}$. Here, $s_{\mathfrak{ne}}^{(u-1)}$ is the shared message from node $\mathfrak{n}$. Otherwise, it calculates $\hat{f}_{\mathfrak{en}}^{u}$ with only $\tilde{f}_{\mathfrak{en}}^{u}$. $f_{\mathfrak{e}}^{u}$ is the variation for the update of node $\mathfrak{e}$. With this value, node $\mathfrak{e}$ controls its own carrier frequency. Finally, $s_{\mathfrak{en}}^{u}$ is the shared message from node $\mathfrak{e}$ to node $\mathfrak{n}$.

Algorithm 3. Distributed frequency synchronization.
for every $n \in \chi$ **do**
Calculate $\tilde{f}_{\mathfrak{en}}^{u}=\frac{1}{R_{\mathfrak{n}}}\sum_{r=1}^{R_{\mathfrak{n}}}\tilde{f}_{\mathfrak{en}}^{u}(r)$
Calculate $\hat{f}_{\mathfrak{en}}^{u}=\frac{1}{2}\left\{\tilde{f}_{\mathfrak{en}}^{u}+\left(s_{\mathfrak{ne}}^{(u-1)}-f_{\mathfrak{e}}^{(u-1)}\right)\right\}$
end **for**
Calculate $f_{\mathfrak{e}}^{u}=\frac{1}{O_{\mathfrak{e}}+1}\sum_{n\in\chi}\hat{f}_{\mathfrak{en}}^{u}$
for every $n \in \chi$ do
Calculate $s_{\mathfrak{en}}^{u}=f_{\mathfrak{e}}^{u}-\hat{f}_{\mathfrak{en}}^{u}$
end **for**

IV. Simulation Results

For the network construction, three types of node distributions are considered according to the node-density. In the first distribution, the total number of nodes (N_{t}) in the network is 16, and the average number of one-hop neighbor nodes ($O_{\mathfrak{e}}$) is about three. In the second and third distributions, the total number of nodes and the average number of one-hop neighbor nodes in the network are (28 and 4) and (54 and 8), respectively. For a simple model of a backhaul scenario, we assume a one-hop coverage of 10 km, and the maximum propagation delay (PD_{M}) is then 10/(3×10^5) = 33.3 μs. Randomly distributed nodes each update their T, *FFT*, and carrier frequency every update period. In the simulation taking

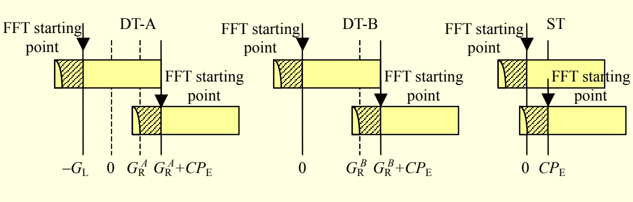

Fig. 6. Three types of time synchronization schemes: DT-A, DT-B, and ST.

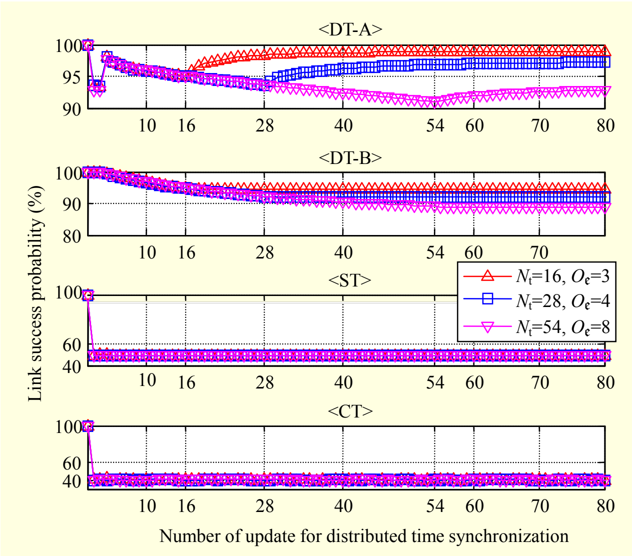

Fig. 7. Link success probability of time synchronization for DT-A, DT-B, ST, and CT under fixed scenario.

a long coverage into account, the preamble signal and ranging code in IEEE 802.16m, which are suitable for long coverage among conventional standards, are used for an estimation of t/F_{off} [15], [24].

As shown in Fig. 6 for the distributed time synchronization, three kinds of algorithms are compared for a more detailed and DT-B indicates that only the *FFT* with the fixed T (T_{rf}) is adjusted according to Algorithm 1 with the same overhead of DT-A, $G_L+G_R^A=G_R^B$. For a performance comparison, ST controls the *FFT* within CP_E according to the maximum received signal to interference and noise ratio (SINR) without a guard interval. The SINR can be calculated through a numerical analysis with the assumption of the known receiving times and power of each transmitted node. A similar method was applied in [20]. Based on IEEE 802.16m, CP is $1/4T_s$ = 22.8 μs, where T_s = 91.4 μs denotes an OFDM symbol length. CP_M = 2.5 μs based on Pedestrian channel B [25], CP_E = 20.3 μs, $G_L = G_R^A$ = 11.4 μs by the assumption of $CP/2$, and G_R^B = 22.8 μs. ΔT_{cr} and ΔFFT_{cr} are set to 1 μs.

In the fixed scenario, we drop 16, 28, and 54 nodes randomly in the 40 km × 40 km square simulation plane because the performance of the time synchronization depends on the relationship between the total number of nodes, *CP*, and one-hop coverage. In other words, if the ratio between one-hop coverage and *CP* length according to the total number of nodes is similar to the parameters described above, similar performances can be obtained. All the positions of the nodes are fixed after dropping them. Figure 7 shows the system level performances according to the node distribution in the fixed scenario. In the figure, CT is used for a comparison of the time synchronization in WSNs [14]. This is a method used for setting the same T for all nodes. In addition, the link success probability is an indicator for the time synchronization between nodes. There is one link for a pair of any two nodes, and if both nodes are synchronized with each other, the link success probability is 1. The performance shows the link success probability for all links between multiple pairs in the network. A node per each update time is joined to the network. All of the schemes converge in about 20 updates after all nodes are joined. DT-A has the best performance among the three schemes, which approaches a 92% to 99% link success probability. This is because DT-A exploits both request messages and the guard interval. In addition, DT-A shows a relatively dynamic change in performance because it takes an active part in synchronization with changing the *FFT*, as well as T, compared with other schemes. In addition, for example, all 16 nodes are joined when the update time is 16. The DT-A scheme for all nodes is then applied. Thus, the link success probability is increased after an update time of 16 when the total number of nodes is 16. Similar results are presented for 28 nodes and 54 nodes. DT-B also achieves an 89% to 95% link success probability. DT-B has a worse performance than DT-A despite the same overhead. This is because the transmission setting procedure is not performed. We notice that the performance of ST is poor, and there is no performance improvement regardless of the update, although it has a relatively low overhead because of using only CP_E. In addition, the performance of CT is worse than that of ST. This is because CT concentrates on the synchronization T, not t, and the WSNs mainly consider a short coverage.

If we use CP_{long} to obtain a 100% link success probability for ST, the required overhead (ς) compared with DT-A and DT-B is then calculated as follows:

$$\varsigma = \left(CP_{\text{long}} - CP\right)\cdot N_{\text{sym}} - (G_L + G_R^A), \tag{15}$$

where N_{sym} is the number of symbols per fr/sfr. For example, if CP_{long} is $1/2T_s$= 45.71 μs, which is over $PD_M + CP_M$ = 35.8 μs, and one fr/sfr contains eight symbols, the additional overhead per fr/sfr is then 160.48 μs. This value is more than one OFDM symbol length and, moreover, can be increased more when increasing the number of symbols per fr/sfr, while we can reduce the amount of overhead using the proposed distributed

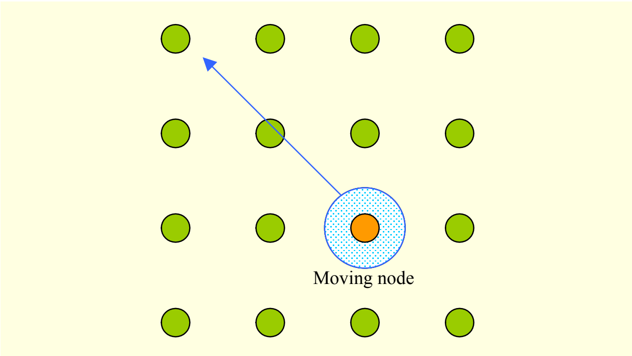

Fig. 8. Movement scenario for distributed time synchronization.

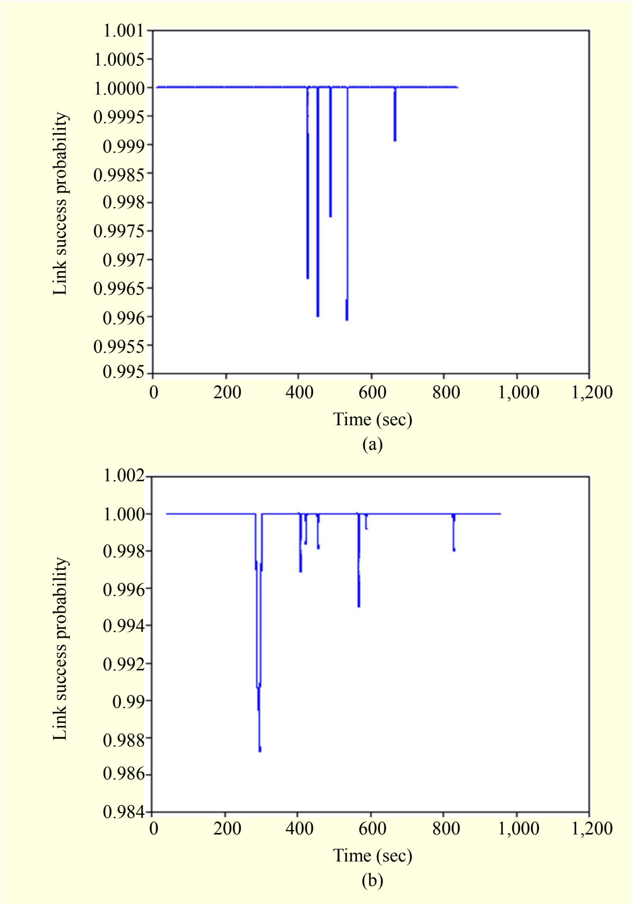

Fig. 9. Link success probability for distributed time synchronization with DT-A in movement scenario: (a) 60 km/h; (b) 250 km/h.

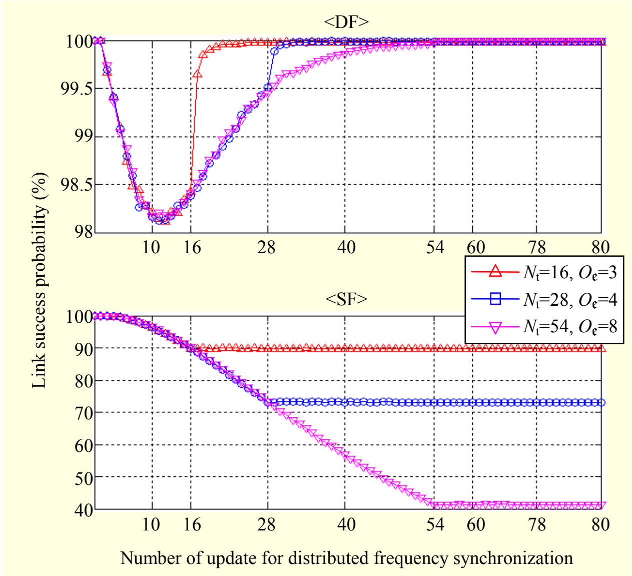

Fig. 10. Link success probability of frequency synchronization with DF and SF in fixed scenario; N_t, total number of nodes; and O_e, average number of one-hop neighbor nodes.

time synchronization method.

In addition, we simulate the link success probability of DT-A using an OPNET simulator for the moving scenario. In the moving scenario, 16 nodes are allocated to the grid, and one node moves through fixed nodes, as shown in Fig. 8. We consider two moving scenarios: 60 km/h and 250 km/h. The link success probabilities of the two cases are presented in Fig. 9. In the simulation, the update interval is 100 ms, which is the time duration of five superframes in IEEE 802.16m. In Fig. 9(a), the average link success probability is 99.99% and the minimum performance is 99.6% at the duration of 400 s to 600 s. This is because the moving node at that time is located in the center of the simulation plane, and there are many neighbor nodes for the time synchronization. In Fig. 9(b), the average and minimum link success probabilities are 99.97% and 98.72%, respectively. The minimum performance is degraded compared with that of 60 km/h. This is because the node moves fast before the completion of the gradual time synchronization with the neighbor nodes.

On the other hand, Fig. 10 shows the gain of the distributed frequency synchronization in the fixed scenario. R_n is set to 5 for the simulation. We define the link success when it synchronizes not only a target node but also neighbor nodes within less than 1% subcarrier spacing. The 1% threshold value is reasonable enough in terms of the bit error rate [26]. One node per each update time is joined to the network until all nodes are joined. In the figure, DF denotes the distributed frequency synchronization scheme of Algorithm 3, and SF for a performance comparison denotes that each node synchronizes with only its sponsor node, similar to that in cellular systems [15]. In addition, five offset estimations are also applied for SF. DF has a link success probability of over 99.9%, while the minimum performance of SF is 40%. This is because SF depends on only one sponsor node without considering the neighbor nodes, and the estimation error with the sponsor node is not corrected.

V. Conclusion

In this paper, we proposed a distributed synchronization method and analyzed its performance in terms of the link success probability. The distributed time synchronization provides time synchronization with one-hop neighbor nodes by using request messages and the guard interval without a long *CP*. In addition, the distributed frequency synchronization builds the global frequency synchronization using the average value and sharing messages. Simulation results show that the proposed method outperforms the conventional scheme by the cooperation of neighbor nodes and completes the synchronization gradually in OFDMA-based WMNs.

References

[1] O. Tipmongkolsilp, S. Zaghloul, and A. Jukan, "The Evolution of Cellular Backhaul Technologies: Current Issues and Future Trends," *IEEE Commun. Surveys Tutorials*, vol. 13, no. 1, First Quarter 2011, pp. 97-113.

[2] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless Mesh Networks Design — A Survey," *IEEE Commun. Surveys Tutorials*, vol. 14, no. 2, Second Quarter 2012, pp. 299-310.

[3] G. Aggelou, "Wireless Mesh Communication Architectures and Protocols," *Wireless Mesh Networking*, Punta Gorda, FL: McGraw-Hill Professional, 2008, Chapter 2.

[4] P802.11-REVmb/D12, *Nov 2011 — IEEE Standard for Information Technology — Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks — Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Mar. 2012.

[5] 802.16-2004, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, Nov. 2004.

[6] 802.15.5-2009, *IEEE Recommended Practice for Information Technology — Telecommunications and Information Exchange between Systems — Local and Metropolitan Area Networks — Specific Requirements Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs)*, May 2009.

[7] I.F. Akyildiz et al., "Wireless Sensor Networks: A Survey," *Comput. Netw.*, vol. 38, no. 4, Mar. 2002, pp. 393-422.

[8] H. Kim et al., "The Trend Analysis for Wireless Mesh Networks," *Nat. IT Ind. Promotion Agency*, vol. 1479, Jan. 2011, pp. 1-19.

[9] H.J. Kwon et al., "Geralized CSMA/CA for OFDMA Systems: Protocol Design, Throughput Analysis, and Implementation Issues," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, Aug. 2009, pp. 4176-4187.

[10] R.V. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Norwood, MA: Artech House, 2000.

[11] K.A. Hamdi, "Precise Interference Analysis of OFDMA Time-Asynchronous Wireless Ad-hoc Networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, Jan. 2010, pp. 134-144.

[12] C.H. Park et al., "An Analysis of TDoA Effect for OFDMA-Based Wireless Mesh Networks," *IEEE Int. Conf. Commun.*, Kyoto, Japan, June 5-9, 2011, pp. 1-4.

[13] O. Simeone et al., "Distributed Synchronization in Wireless Networks," *IEEE Signal Process. Mag.*, vol. 25, no. 5, Sept. 2008, pp. 81-97.

[14] M.K. Maggs, S.G. O'Keefe, and D.V. Thiel, "Consensus Clock Synchronization for Wireless Sensor Networks," *IEEE Sensors J.*, vol. 12, no. 6, June 2012, pp. 2269-2277.

[15] 802.16m-2011, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface*, May 2011.

[16] 3GPP TS 36.211 v8.9.0, *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRAN)*; *Physical Channels and Modulation (Release 8)*, Dec. 2009.

[17] S.G. Lee and X. Ma, "Symbol Detection on Asynchronous OFDMA Mesh Networks with Timing Misalignment," *Military Commun. Conf.*, Baltimore, MD, USA, Nov. 7-10, 2011, pp. 2188-2193.

[18] S.G. Lee and X. Ma, "Timing Adjustment Techniques to Mitigate Interference between Multiple Nodes in OFDMA Mesh Networks," *IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 3508-3511.

[19] C.H. Park et al., "A Bidirectional Successive Detection Technique for Asynchronous OFDMA-Based Wireless Mesh Networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 7, Sept. 2012, pp. 3346-3352.

[20] C.H. Park et al., "An MMSE-BSD Technique for Wireless Mesh Networks with TDoAs," *IEEE Int. Conf. Signal Process., Commun. Comput.*, Hong Kong, China, Aug. 12-15, 2012, pp. 43-46.

[21] J. Kim, K.J. Lim, and M. Lee, "Gradual Time Synchronization for Wireless Mesh Networks Based on OFDMA," *7th Int. Conf. Next Generation Mobile Apps., Services, Technol.*, Prague, Czech Republic, Sept. 25-27, 2013, pp. 187-191.

[22] P.H. Dana, "Global Positioning System (GPS) Time Dissemination for Real-Time Applications," *Real-Time Syst, Int. J. Time Critical Comput. Syst.*, vol. 12, no. 1, Jan. 1997, pp. 9-40.

[23] J.H. Kim et al., "Distributed Frequency Synchronization for Global Synchronization in Wireless Mesh Networks," *Int. Conf. Netw. Wireless Commun.*, Kuala Lumpur, Malaysia, Oct. 8-10, 2012, pp. 652- 656.

[24] J. Kim et al., "Performance Analysis of Synchronization for an OFDMA System," *Int. Conf. Wireless Mobile Commun.*, Venice, Italy, June 24-29, 2012, pp. 376-380.

[25] ITU-R M.1225, *Guidelines for Evaluations of Radio*

Transmission Technologies for IMT-2000, ITU-R, 1997.
[26] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Weiner Phase Noise," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, Feb./Mar./Apr. 1995, pp. 191-193.

Jihyung Kim received his BS, MS, and PhD from the department of electrical and electronic engineering at Yonsei University, Seoul, Rep. of Korea, in 2000, 2002, and 2007. In February 2007, he joined ETRI, Daejeon, Rep. of Korea. His current research interests include distributed architectures in wireless communications.

Jung-Hyun Kim received his BS and MS degrees from the department of electronic and electrical engineering at Yonsei University in 2006 and 2008, respectively. From 2010 to 2012, he was a member of the engineering staff at ETRI, where he contributed to a team developing fundamental technology on next-generation adaptive wireless mesh communication systems for on-the-move nodes. He is currently a PhD candidate in the department of electrical and electronic engineering at Yonsei University. His research interests include application of OFDMA systems, error correcting codes, network coding, distributed storage systems, and compressed sensing.

Kwangjae Lim received his BS, MS, and PhD from the department of electronics engineering at Inha University, Incheon, Rep. of Korea, in 1992, 1994, and 1999. In March 1999, he joined ETRI, Daejeon, Rep. of Korea. Since 1999, he has worked on the standardization of mobile and satellite communications. Since 2010, he has headed a mobile application team. His research interests are in mobile and wireless communications.