Vertex Coloring based Slot Reuse Scheduling for Power Line Communications

Sung-Guk Yoon[†]

Abstract – Power line communication (PLC) is one of the major communication technologies in smart grid since it combines good communication capability with easy and simple deployment. As a power network can be modeled as a graph, we propose a vertex coloring based slot reuse scheduling in the time division multiple access (TDMA) period for PLCs. Our objective is to minimize the number of assigned time slots, while satisfying the quality of service (QoS) requirement of each station. Since the scheduling problem is NP-hard, we propose an efficient heuristic scheduling, which consists of repeated vertex coloring and slot reuse improvement algorithms. The simulation results confirm that the proposed algorithm significantly reduces the total number of time slots.

Keywords: Conflict graph, Power line communications, Scheduling, Vertex coloring algorithm

1. Introduction

Because of the ubiquitous characteristics of power-line communication (PLC) for power grids, PLC takes on an important role for the next generation power grid, i.e., *smart grid*, which requires two way communications. According to the operational bandwidth, PLC systems can be classified into narrowband and broadband ones. Narrowband PLC uses sub 500 kHz and its maximum achievable throughput is up to several hundred kbps. On the other hand, broadband PLC runs at 1.8 MHz-250 MHz and the state-of-the-art broadband PLC standard can have up to Gbps throughput performance with shorter transmission ranges [1]. A recent smart grid project in Spain uses both narrowband PLC and broadband PLC. They are used for the connections between the substations and between the substations and smart meters, respectively [2, 3].

PLC has both a time division multiple access (TDMA) period to guarantee quality of service (QoS) and carrier sense multiple access with collision avoidance (CSMA/CA) period for impromptu data or control packet transmissions [4]. The TDMA period with proper time slot allocation between PLC stations has the advantage of exclusive channel access without having interference from other PLC stations. Therefore, scheduling, i.e., which station uses which time slot, is very important in the TDMA period.

Many researches on PLC scheduling in the TDMA period have recently been conducted. A simple heuristic time slot allocation scheme based on an interference matrix was proposed in [5]. This heuristic scheme is not scalable, i.e. it can only be used for two neighboring PLC networks. In [6, 7], a fair time slot assignment scheme was proposed for PLC in a multi-cell distribution network. However, this

work does not consider slot reuse[1]. Lee et al. proposed another time slot assignment scheme for routing in PLC access networks [8]. They formulated an optimization problem and proposed a heuristic algorithm. In [9], TDMA scheduling was proposed for tree based PLC networks. However, PLC in distribution networks forms diverse kinds of topologies including tree and mesh topologies.

A vertex coloring algorithm [10, 11] can be mapped onto the scheduling algorithm when the colors are regarded as time slots in scheduling. Several time and frequency scheduling schemes based on vertex coloring have been proposed for wireless communications [12]. To the best of our knowledge, no vertex coloring based scheduling research has been carried out for PLC. Although vertex coloring algorithm is NP-hard, there are several efficient heuristic algorithms which give suboptimal solutions in a polynomial time with bounded performance gap [10].

In this paper, we propose a slot reuse TDMA scheduling for PLC networks using a vertex coloring algorithm. The proposed scheduling is not limited to the tree structure and uplink transmissions. We first create a network graph from a real PLC network topology. Then, the network graph is transformed into a conflict graph, and the scheduling problem is the same as the vertex coloring problem in the conflict graph. Since the vertex coloring problem is NP-hard, we use a greedy heuristic algorithm to get a suboptimal solution [11]. To satisfy the various QoS requirements of each station, the vertex coloring algorithm is repeatedly executed. In addition, we also propose a slot reuse improvement (SRI) algorithm that searches the conflict graph in an attempt to reuse time slots.

---

[1] Typically, only one station is scheduled in a time slot for TDMA period. However, if another station is located in out of the transmission range of the scheduled station, it can be also scheduled in the time slot. We call it as a slot reuse.

† Corresponding Author: School of Electrical Engineering, Soongsil University, Korea. (sgyoon@ssu.ac.kr)

The simulation results show that the proposed scheduling significantly improves the slot reuse, resulting in a shorter TDMA period to fulfill the QoS requirement of each station.

The remainder of this paper is organized as follows. In Section 2, we describe our system model for a PLC network. O proposed slot reuse TDMA scheduling protocol is described in Section 3. After evaluating the proposed protocol in Section 4, we conclude our paper in Section 5.

2. System Model

We consider a homogeneous PLC network in which all of the PLC stations use the same PLC protocol, such as IEEE 1901[2] [13]. A beacon interval, which is synchronized with the AC line cycle, consists of a beacon period, a QoS guaranteed TDMA period, and a CSMA/CA period, which is for impromptu and best effort data or control packet transmissions. Fig. 1 shows an example of the beacon period. One beacon period of IEEE 1901 lies on two AC line cycles and its length is 33.33 msec for US and Korea and 40 msec for Europe.
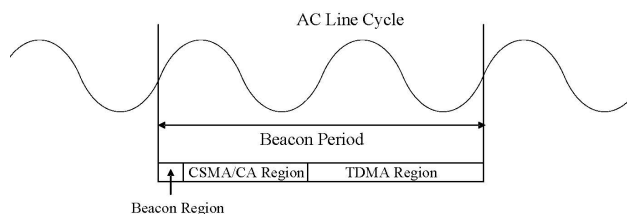

Fig. 1. Example of IEEE 1901 beacon period. The beacon periods are synchronized with the AC line cycle for robust transmission. A beacon period consists of three regions: the beacon, CSMA/CA, and TDMA regions.
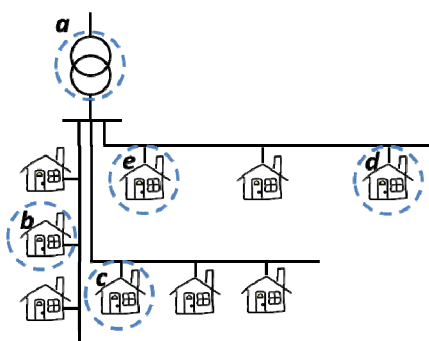

Fig. 2. Example of a PLC topology. There are three branches originating from one substation and three houses per branch. Among them, the substation and four houses have PLC based smart meters.
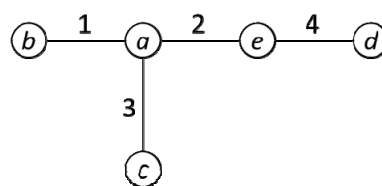

Fig. 3. Network graph for the example topology. The nodes and links stand for the stations and communication channels, respectively.
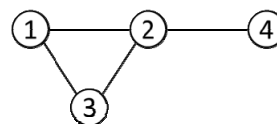

Fig. 4. Conflict graph for the example topology. The vertices and edges mean the communication links and conflict relationships, respectively.

We consider a PLC network that consists of *N* stations within the same distribution network, as shown in Fig. 2. In this example, there are five PLC stations. That is, four out of the nine houses and the substation have PLC based smart meters. It is assumed that the PLC signal reaches up to the second neighbor houses because of the signal attenuation.

2.1 Graph model

The actual PLC topology shown in Fig. 2 can be changed into a logical network graph as shown in Fig. 3. Let $G = (V, E)$ denote the network graph, where V is the set of nodes, which are the PLC stations, and E is the set of links, which constitute the connecting links formed by those stations that are within transmission range of each other. We assume that all of the links in the network graph are bi-directional, i.e., channel reciprocity. In this example, *a* can communicate with *b*, *c*, and *e*; and *d* can only communicate with *e*. Note that we use the term station for a PLC module installed in a house and node for the network graph.

From the network graph, we can also make a conflict graph which models the interference relationship [14]. Let $G_C = (V_C, E_C)$ denote the conflict graph, where V_C and E_C are the set of vertices and edges, respectively[3]. A vertex and edge in the conflict graph correspond to a link in the network graph and interference relationship, respectively. Fig. 4 shows the conflict graph generated by the network graph shown in Fig. 3. When *a* is communicating with *b* (vertex *1*), the node cannot communicate with *c* (vertices *3*) and *e* (vertices *2*), so there are edges between vertices *1*, *2*, and *3*. However, *e* and *d* (vertex *4*) can communicate with each other during the communication between *a* and

---

b (vertex *1*), resulting in there being no edge between *1* and *4*.

Note that the conflict graph makes it easier to understand the interference relationship between the stations, while the network graph clarifies the actual topology.

2.2 Scheduling model

To express the scheduling problem, we define several variables. We first define a time slot allocation indicator $s_{v,t}$ for a vertex $v \in V_C$ and time slot $t \in \mathbb{T}$ as

$$s_{v,t} = \begin{cases} 1, & \text{if } t \text{ is allocated to } v \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathbb{T}$ denotes a set of time slots, i.e., $\mathbb{T} = \{1, 2, \cdots, T\}$ and T is the maximum scheduling time slot. The time slot allocation vector for v is defined as $\vec{s_v} = [s_{v,1}, s_{v,2}, \cdots, s_{v,T}]$. We finally define the whole time slot allocation vector $\vec{s} = [\vec{s_v}]$, $v \in V_C$. In this terminology, TDMA scheduling involves assigning 0 and 1 to $s_{v,t} \forall v, t$.

If vertices *1* and *2* in Fig. 4 are scheduled in the same time slot the two transmissions will fail due to a severe interference. To avoid this, the interference constraint should be considered. We can express the interference constraint as

$$s_{v,t} + s_{v',t} \le 1, \text{ if } \exists l \in E_C \text{ between } v \text{ and } v'. \tag{2}$$

This interference constraint allows only one vertex to be activated among its neighbor vertices.

Lastly, to represent the QoS requirement of each station, we define a utility function $U_v(\vec{s_v})$ which denotes the degree of satisfaction for v when v is scheduled as $\vec{s_v}$. One simple utility function counts the number of assigned time slots for the station. It can be expressed as[4]

$$U_v(\vec{s_v}) = \sum_{t \in \mathbb{T}} s_{v,t} . \tag{3}$$

Then, a QoS requirement constraint of v can be expressed as $U_v(\vec{s_v}) \ge U_v$ where U_v denotes the QoS requirement of v. Note that each station can have a different utility function.

We assume that a central station, which takes charge of the scheduling role, has information on the topology and utility function for each station. In Fig. 2, *a* is the central station. Although *d* cannot directly communicate with *a*, *d* can obtain the scheduling information through proxy beacon[5].

---

[4] We use the utility function in this paper.
[5] According to the IEEE 1901 specification, the scheduling information is contained in beacon. Some stations that cannot receive beacon from the central station can receive the information by proxy beacon. Intermediate stations between the central station and the hidden stations rebroadcast

3. Slot Reuse TDMA Scheduling for PLC

In this section, we build a slot reuse TDMA scheduling which minimizes the number of allocated time slots, while satisfying the QoS requirements of all of the stations.

In IEEE 1901, the length of the TDMA and CSMA / CA periods is adjustable according to the traffic requirement. For instance, when many stations want to use QoS guaranteed services, such as for distribution system monitoring, it requires a longer TDMA period at a cost of a shortened CSMA / CA period. Therefore, efficient resource allocation in the TDMA period expands the length of the CSMA/CA period.

We can formally define the scheduling problem as

$$(\mathbf{P}) \quad \underset{s_{v,t}}{\text{minimize}} \quad |\mathbb{T}|$$

$$\text{subject to } s_{v,t} \in \{0,1\} \quad \forall v \in V_C, \ t \in \mathbb{T}$$

$$s_{v,t} + s_{v',t} \le 1, \text{ if } \exists l \in E_c \text{ between } v \text{ and } v'$$

$$U_v(\vec{s_v}) \ge U_v, \forall v \in V_C .$$

The first constraint of (**P**) represents the scheduling variable $s_{v,t}$ can only be a binary integer, and the second and third constraints are the interference constraint and QoS constraint, respectively. Problem (**P**) is a linear integer programming, which is NP-hard. Therefore, we propose a heuristic scheduling to obtain a suboptimal solution.

3.1 Homogeneous unit QoS requirement

We first consider a special case, where each station has a unit QoS requirement. That is, the third constraint is changed into

$$\sum_{t \in \mathbb{T}} s_{v,t} \ge 1, \quad \forall v \in V_C \tag{4}$$

Problem (**P**) with the new constraint is exactly the same as a vertex coloring problem if the time slots are regarded as colors. The vertex coloring problem is a color assignment problem to each vertex with the minimum number of colors, while no two adjacent vertices can have the same color. This problem of finding the minimum number of colors is also NP-hard [10].

To get a suboptimal solution in a polynomial time, we use a recursive largest first (RLF) algorithm [11], which is a kind of greedy algorithm with maximum $O(N^3)$ complexity, where N is the number of vertices in a graph, i.e., the number of PLC stations in this paper. This algorithm uses at most $d+1$ colors where d is the maximum degree of all of the vertices. A degree of a vertex means the number of neighboring vertices of the vertex, i.e., the number of edges of the vertex.

---

proxy beacon right after receiving beacon [13].

Algorithm 1: RLF Vertex Coloring Algorithm

1: Input: $G_C = (V_C, E_C)$
2: // Initialization
3: $C = \varnothing$;   // set of colors
4: $W = V_C$;   // set of uncolored vertices
5: // Coloring Loop
6: while ($|W| > 0$) { // until all vertices are colored.
7:   select a vertex $x \in W$ that has the maximal degree;
8: $C = C \cup \{c\}$;   // add a new color c
9: $F(x) = \{c\}$;   // this color is assigned to x
10: $W = W \setminus \{x\}$; // remove x from W
11: $SNN(x)$ = set of non-neighbors for x;
12: while ($|SNN(x)| > 0$) {
13:   Find $y \in SNN(x)$ that has maximum number of common neighbor of x;
14: $F(y) = \{c\}$;   // assign the same color to y
15: $SN(y)$ = set of neighbors for y;
16: $SNN(x) = SNN(x) \setminus \{\{y\} \cup SN(y)\}$;
17: $W = W \setminus \{y\}$;
18: }
19: }
20: Output: $F(x), \quad \forall x \in V_C$

Algorithm 1 shows the RLF vertex coloring algorithm. The algorithm selects a vertex *x* with the largest degree. Then, it assigns a color to the vertex and removes it from the set of uncolored vertices *W*. The color is also painted to a vertex *y* that is not a neighbor of *x*, and *y* is also removed from *W*. This procedure is repeated until there is no vertex to paint with the color. Then, the algorithm assigns a new color to the vertices through the same algorithm. This continues until all of the vertices have been assigned colors.

Table 1. Scheduling result for example topology with unit QoS requirement.

Vertex ID \ Time slot	1	2	3
1 (U_1=1)	0	1	0
2 (U_2=1)	1	0	0
3 (U_3=1)	0	0	1
4 (U_4=1)	0	1	0

Table 1 shows the scheduling result for the RLF vertex coloring algorithm. If it is 1 the time slot is allocated to the vertex. Otherwise, it is 0. In this example, since vertex *2* has the largest degree, it is assigned time slot 1 and no other vertices are assigned this time slot. In time slot 2, since vertices *1* and *4* have no edge, these two vertices are assigned to this time slot.

3.2 Heterogeneous QoS requirement

We then consider the general case where each station has its own QoS requirement. We propose a repeated vertex coloring algorithm and SRI algorithm for the general case.

Repeated vertex coloring algorithm: When a station has more than a unit QoS requirement, it requires more than one time slot. To this end, we propose the repeated vertex coloring algorithm.

Algorithm 2: Repeated Vertex Coloring Algorithm

1: // Initialization
2: Generate a conflict graph G_C from a network graph G
3: Initialize QoS requirement.
4: // repeatedly execute vertex coloring algorithm
5: while(all constraints are fulfilled) {
6:   Regenerate G_C // by removing vertices whose $U_v = 0$
7:   Proceed *RLF Vertex Coloring Algorithm(G_C)*
8:   For all $v \in V_C$ {   // update QoS requirement
9: $U_v = U_v - 1$;
10: }
11: }

The repeated vertex coloring algorithm is shown in Algorithm 2. As a result of the RLF vertex coloring algorithm, all of the stations are assigned a single time slot. Therefore, the QoS requirement of each station is decreased by one (lines 8-10) and the QoS requirements of some of the stations may be fulfilled, i.e. $U_v = 0$. The algorithm regenerates the conflict graph by removing the vertices and edges connected with the vertices (line 6). This procedure is repeatedly executed until the QoS requirements of all of the stations are satisfied.

Table 2. Scheduling result of repeated vertex coloring algorithm for the same topology with U_1=2, U_2=1, U_3=3, and U_4=4.

Vertex ID \ Time slot	1	2	3	4	5	6	7
1 (U_1=2)	0	1	0	1	0	0	0
2 (U_2=1)	1	0	0	0	0	0	0
3 (U_3=3)	0	0	1	0	1	1	0
4 (U_4=4)	0	1	0	1	0	1	1

Table 2 shows the results of the repeated vertex coloring algorithm for the same topology with QoS constraints of U_1=2, U_2=1, U_3=3, and U_4=4. Since this algorithm starts with the RLF vertex coloring algorithm, the first three time slots in Table 2 are the same as those in Table 1. The QoS requirement of vertex *2* is fulfilled after the first coloring algorithm is executed, so it is not scheduled hereafter. Next, the RLF vertex coloring algorithm is executed with the updated graph whose vertices are {*1*, *3*, *4*}, and then its result is time slots 4 and 5. The procedure continues in a similar manner until the QoS constraints of all of the vertices are fulfilled. The RLF vertex coloring algorithm is executed four times in this example.

Note that since the RLF vertex coloring algorithm assigns a single time slot to each station, the RLF vertex coloring algorithm is executed the same number of times as the maximum QoS constraint among the stations. In this example, since a maximum QoS requirement is four

(vertex *4*), the RLF vertex coloring algorithm runs four times.

Repeated vertex coloring algorithm with SRI: The vertex coloring algorithm only aims to color all of the vertices with the minimum number of colors, so it does not consider the case where a vertex can have two colors. For example, in Table 2, vertex *4* can be scheduled in time slot *3* without interfering with vertex *3*, as shown in Fig. 4. To reuse the time slot, we propose an SRI algorithm. The SRI algorithm runs after the RLF vertex coloring algorithm. It looks at all of the vertices in an attempt to reuse the time slots.

The proposed repeated vertex coloring algorithm with SRI is shown in Algorithm 3. The SRI algorithm follows the RLF vertex coloring algorithm (lines 11-23). For each color *c*, the algorithm initializes all of the vertices as unmarked. Then, it marks those vertices that use *c* and their neighbor vertices. Among the unmarked vertices, the SRI algorithm additionally assigns the given color *c* to the vertex that has the maximum U_v. SRI again marks the vertex and its neighbor vertices until all of the vertices are marked.

Algorithm 3: Repeated Vertex Coloring Algorithm with SRI

1: // Initialization
2: Generate a conflict graph G_C from a network graph G
3: Initialize QoS requirement.
4: // repeatedly execute vertex coloring algorithm
5: While(all constraints are fulfilled) {
6:　Regenerate G_C // by removing vertices whose $U_v = 0$
7:　Proceed *RLF Vertex Coloring Algorithm(G_C)*
8:　For all $v \in G_C$ {
9:　　$U_v = U_v - 1$;
10:　}
11: // SRI algorithm
12: For all $c \in C$ {
13:　Unmark all vertices
14:　Mark vertices x that $F(x) = c$;
15:　Mark vertices that has an edge with x
16:　For all unmarked v {
17:　　Choose y which has the maximum U_v
18:　　$F(y) = F(y) \cup \{c\}$;
19:　　Mark the vertex y
20:　　Mark vertices that has an edge with y
21:　　$U_y = U_y - 1$;
22:　}
23: }
24: }

Table 3 shows an example of scheduling for the repeated coloring algorithm with SRI under the same QoS constraints as those in Table 2. The reused time slots due to the SRI algorithm are in bold, i.e., time slots 3 and 5 for vertex *4*. In this example, to fulfill all of the QoS requirements, the

Table 3. Scheduling result of repeated vertex coloring algorithm with SRI for the same topology and QoS requirements.

Time slot Vertex ID	1	2	3	4	5	6
1 (U_1=2)	0	1	0	1	0	0
2 (U_2=1)	1	0	0	0	0	0
3 (U_3=3)	0	0	1	0	1	1
4 (U_4=4)	0	1	**1**	1	**1**	0

repeated coloring algorithm with SRI and without SRI takes 6 and 7 time slots, respectively.

4. Evaluations

In this section, we compare the performance of the proposed slot reuse TDMA scheduling to those of no reuse scheduling and tree-based scheduling [9] in terms of scheduling the finish time.

4.1 Simulation settings

We consider three PLC topologies. Topology-1 is the one shown in Fig. 2. In Topology-2, there are two cells, each of which has five PLC stations. Within a cell, all of the stations can communicate with each other, but only one station in cell 1 and two stations in cell 2 can communicate with the other cell. Similarly to Topology-1, in Topology-3, there are five power line branches originating from a substation and four houses in each branch. PLC based smart meters are installed at all houses and a PLC module is also installed at the substation. This is a typical advanced metering infrastructure (AMI) in a distribution network [15]. We assume that the PLC signals can reach their two hop neighbors in Topology-3.

Table 4. IEEE 1901 timing parameters

Parameters	Value
T_{PHY}	110.48 usec
T_{sym}	48.52 usec
T_{RIFS}	120 usec
T_{ACK}	110.48 usec
T_{AIFS}	30 usec

It is assumed that one time slot consists of a physical layer (PHY) header, one OFDM data symbol, response interframe space (RIFS), acknowledgement (ACK) frame transmission, and allocation interframe space (AIFS). The period of one time slot is

$$T_{slot} = T_{PHY} + T_{sym} + T_{RIFS} + T_{ACK} + T_{AIFS} . \qquad (5)$$

The timing parameters for the IEEE 1901 TDMA protocol are given in Table 4 [13] and $T_{slot} = 419.48$ usec.
We assume that the channel does not fluctuate much

during a beacon period. The QoS requirement is defined as the number of time slots in one beacon period. The QoS requirement of each station is randomly chosen between 1 and 5 time slots[6]. To obtain the results, simulations are performed 100 000 times and the results are averaged.

Note that in our evaluations, we do not consider packet transmission errors. This is because time slots are assigned to each station without having severe interference and because it is assumed that a proper modulation and coding scheme is applied.

4.2 Simulation results

The proposed algorithm firstly generates the conflict graph from the network graph. Table 5 lists the generated graphs. According to the definition of the conflict graph, the number of links in the network graph is the same as the number of vertices in the conflict graph. In Topology-1, the numbers of vertices and edges are the same. This means that the graph is sparsely connected. The ratios of the number of vertices to the number of edges in the other two graphs are 3.5 and 4.44, respectively. That is, Topology-3 is the most connected graph among the three topologies.

Fig. 5 shows the number of allocated time slots required to fulfill the QoS requirements of all of the stations. With no reuse scheduling, only one station can send its data in each time slot. In the case of Topology-3, to fulfill the QoS

Table 5. Generated graphs

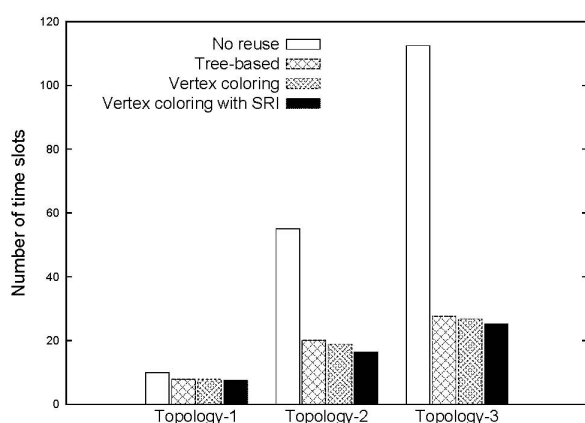|	Network graph	Conflict graph
Topology-1	(5, 4)	(4, 4)
Topology-2	(10, 22)	(22, 77)
Topology-3	(21, 45)	(45, 200)

Fig. 1. Total number of time slots in the TDMA period to satisfy the QoS requirement of all stations.

requirement of each station, about 112 time slots are needed with no reuse scheduling. Since one time slot takes 419.48 usec, 46.98 msec is needed for the TDMA period, which is longer than one beacon period, i.e., 33.33 msec for US and Korea. Consequentially, no reuse scheduling cannot satisfy the QoS requirement of each station in Topology-3. However, the repeated vertex coloring algorithm with or without SRI only takes about 26 time slots, which is about 10.9 msec, i.e., it affords a 78% improvement. Therefore, with the proposed scheduling, not only are the QoS requirements of all of the stations satisfied in the TDMA period, but also the CSMA/CA period has about 22.43 msec to send the impromptu data or control packets.

The proposed vertex coloring algorithm without SRI slightly outperforms the tree-based scheduling. It is because the tree-based scheduling is optimized for uplink and downlink traffics in the tree topology while the proposed scheme is not limited to the traffics in the tree topology. Therefore, since Topology-1 is the tree topology, the performances for the tree-based scheduling and the proposed vertex coloring algorithm in Topology-1 are the same.

The proposed SRI algorithm further reduces the number of time slots. For Topology-1, Topology-2, and Topology-3, the repeated vertex coloring algorithm with SRI provides improvements of 3.01%, 13.11%, and 5.69%, respectively, compared to those without SRI.

5. Conclusion

We propose a vertex coloring based slot reuse TDMA scheduling for PLC. We formulate the scheduling problem as one of minimizing the number of allocated time slots while satisfying the QoS requirement of each station. This problem is the same as the vertex coloring problem in graph theory, which is NP-hard. Therefore, to solve the problem, a greedy coloring algorithm is used and the algorithm is repeatedly executed to satisfy the QoS requirement of each station. In addition, we also proposed a slot reuse improvement algorithm to further utilize the power line channel. The simulation results showed that the proposed slot reuse TDMA scheduling significantly reduces the total number of allocated time slots.

Acknowledgements

---

[6] When a station simply sends its metering data to the data collection unit, one time slot is enough. On the other hand, when another station sends the voltage/VAR status data as well as metering data, it requires more than one time slot.

References

[1] S. Galli, A. Scaglione, and Z.Wang, "For the Grid and through the Grid: The Role of Power Line Communications in the Smart Grid," *Proc. IEEE*, vol. 99, no. 6, pp. 998-1027, Jun. 2011.

[2] A. Sendin, I. Berganza, A. Arzuaga, A. Pulkkinen, and I. Kim, "Performance Results from 100,000+ PRIME Smart Meters Deployment in Spain," in *Proceedings of IEEE SmartGridComm 2012,* Tainan, Taiwan, Nov. 2012.

[3] A. Sendin, J. Simon, I. Urrutia, and I. Berganza, "PLC Deployment and Architecture for Smart Grid Applications in Iberdrola," in *Proceedings of IEEE ISPLC 2014,* Glasgow, United Kingdom, Mar.-Apr. 2014.

[4] S.-G. Yoon, and S. Bahk, "Adaptive Rate Control and Contention Window Size Adjustment for Power Line Communication," *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 809-816, Apr. 2011.

[5] S. Galli, Y. Masuda, and Y. Urabe, "Time Reuse Algorithms: A Novel Approach to Solving the Issue of Scalability in Dense Power Line Networks," in *Proceedings of IEEE ISPLC 2009,* Dresden, Germany, Mar.-Apr. 2009.

[6] L. P. Do and R. Lehnert, "A Channel Allocation Protocol for Providing Fairness Between Users in Multi-Cell PLC Networks," in *Proceedings of IEEE ISPLC 2011*, Udine, Italy, Apr. 2011.

[7] L.P. Do and R. Lehnert, "Dynamic Resource Allocation Protocol for Large PLC Networks," in *Proceedings of IEEE ISPLC 2012*, Beijing, China, Mar. 2012.

[8] S.S.W. Lee, K.-Y. Li, C.-S. Wu, J.-Y. Pan, and C.-Y. Chuang, "Optimal Bandwidth Guaranteed Routing and Time Slot Assignment for Broadband PLC Access Networks," in *Proceedings of IEEE ISPLC 2012*, Beijing, China, Mar. 2012.

[9] A.M. Sarafi, A.C. Voulkidis, and P.G. Cottis, "Optimal TDMA Scheduling in Tree-Based Power-Line Communication Networks," *IEEE Trans. Power Del.*, vol. 29, no. 5, pp. 2189-2196, Oct. 2014.

[10] J. A. Bondy and U. Murthy, *Graph Theory with Applications*. New York: Elsevier, 1976.

[11] F.T. Leighton, "A Graph Coloring Algorithm for Large Scheduling Problems," *J. Res. Bur. Stand.,* vol. 84, no. 6, pp. 489-506, 1979.

[12] B.-G. Kim, J.-A. Kwon, and J.-W. Lee, "Subchannel Allocation for the OFDMA-Based Femtocell System," *Computer Networks*, vol. 57, no. 17, pp. 3617-3629, Dec. 2013.

[13] IEEE Std P1901-2010 "Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications," 2010.

[14] K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Wireless Networks*, vol. 11, no. 4, pp. 471-487, Jul. 2005.

[15] H. Ferreira, L. Lampe, J. Newbury, and T. Swart, *Power Line Communications: Theory and Applications for Narrowband and Broadband Communications over Power Lines*, John Wiley & Sons, 2010.

Sung-Guk Yoon He received the B.S. and Ph.D. degrees from Seoul National University, Seoul Korea, in 2006 and 2012, respectively. From 2012 to 2014, he was a Postdoctoral Researcher at the same University. He is currently with Sonngsil University as an Assistant Professor since March 2014. His research interests include smart grid and power line communications.