A Routing Algorithm based on Semi-supervised Learning for Cognitive Radio Sensor Networks

Zilong Jin[1], Donghai Guan[1], Jinsung Cho[1] and Ben Lee[2]
[1] *Dept. of Computer Engineering, Kyung Hee University, 446-701, Yongin, Korea*
[2] *School of Electrical Engineering and Computer Science, Oregon State University, 97331, Corvallis, OR, U.S.A.*
{jinzilong, chojs}@khu.ac.kr, donghai@oslab.khu.ac.kr, benl@eecs.orst.edu

Keywords: Machine Learning, Semi-supervised Learning, Routing Algorithm, Cognitive Radio Sensor Networks.

Abstract: In Cognitive Radio Sensor Networks (CRSNs), the cognitive radio technology enables sensor nodes to occupy licensed bands in a opportunistic manner and provides advantages in terms of spectrum utilization and system throughput. This paper proposes a routing scheme based on *semi-supervised learning*, which jointly considers energy efficiency, context-awareness, and optimal path configuration to enhance communication efficiency. A context-aware module is developed to collect and learn context information in an energy-efficient way and a new semi-supervised learning algorithm is proposed to estimate dynamic changes in network environment. A novel routing metric is used to select the most reliable and stable path. Our simulation study shows that the proposed routing algorithm enhances the reliability and stability for CRSNs, and at the same time, significantly improves the packet delivery ratio.

1 INTRODUCTION

In the foreseeable future, tens of billions of electronic devices will be expected to communicate with each other and require a huge amount of radio resources. However, the current radio resources is lack of due to inflexible spectrum sharing rules. In particular, existing wireless networks, such as WLANs, mesh networks, body area networks, and sensor networks, which operate in unlicensed band will suffer from serious spectrum overcrowding problem. The cognitive radio technology that exploits dynamic spectrum sharing techniques is a promising solution to solve this problem (Cesana et al., 2011).

Application of the cognitive radio technology in wireless sensor networks (WSNs) can open up new and unexplored network configuration possibilities and also enable researchers to explore new services. In *Cognitive Radio Sensor Networks* (CRSNs), the sensor devices are capable of sensing a wide spectrum range, dynamically identifying available channels, and intelligently accessing them. Unlike WSNs, CRSNs can operate in licensed bands. *Cognitive Radio Sensor Devices* (CRSDs), which are also referred to as secondary users, share the licensed band with primary users (PUs) who have higher priorities in occupying the bands in a non-interfered manner. This indicates that the topology of CRSN is changing unpredictably due to PUs' activities and causes considerable difficulty in guaranteeing stable and efficient communications (Cesana et al., 2011; Akan et al., 2009; Ali et al., 2011).

There are many excellent prior research focused on the lower layers (PHY/MAC) of cognitive radio technologies. However, routing, which is an important requirement for efficient communication in multi-hop based CRSNs, has not been well explored. Most of prior work on routing solutions are provided in ad-hoc based cognitive radio networks (Sampath et al., 2008; Cheng et al., 2007; Pefkianakis et al., 2008; Wang et al., 2009). These approaches employ spectrum-aware schemes to support routing module in the path selection process. However, these methods do not take into account energy restriction of sensor networks, and thus cannot be directly applied to CRSN. There is only a limited work that focus on routing issues in CRSNs. Parvin and Fujii (2011) and Shah et al. (2013) proposed spectrum-aware routing solutions to guarantee network QoS requirements. In their proposed algorithms, the optimal path is selected to minimize end-to-end delays, but the impact of unpredictable link failure on QoS and communication performance is not considered. In order to dynamically predict the available spectrum resources in CRSN, a machine learning based routing algorithm is proposed in (Yu et al., 2010). The authors employ

Bayesian Learning to estimate the amount of available resources, and then, the most reliable path that contains the largest number of available channels is selected. Unfortunately, this routing scheme is not feasible because a large amount of labeled data, which are used to train their learning algorithm, is difficult to obtain in CRSN.

This paper proposes a new energy efficient routing algorithm that provides reliable communication performance in CRSN. One of the important features of the proposed routing scheme is that it can predict the PUs' influence on spectrum usability and efficiently evaluate link stability. In order to achieve this, a context-aware module was developed to perceive context information such as PUs' activity and varying radio resources. A semi-supervised learning algorithm that can provide good accuracy under limited labeled data is developed to learn the context information, and then predict the available radio sources in the future. Finally, a novel routing metric is defined to indicate end-to-end link stability, and a stable and reliable path is selected based on a semi-Dijkstra algorithm.

The contribution of this paper can be summarized as follows:

- Development of an energy-efficient context-aware module that integrates context information with a context learning function.
- Development of a feasible learning method that can be applied to CRSNs where the available labeled data are limited.
- Derivation of a novel routing metric to provide stable and reliable paths.

The rest of the paper is organized as follows: Section 2 discusses the most relevant related work. Section 3 presents the proposed semi-supervised based routing algorithm. Section 4 validates the performance of our proposed algorithm through extensive simulations. Finally, Section 5 concludes the paper.

2 RELATED WORK

Due to opportunistic channel access nature of the cognitive radio technology, routing techniques for traditional WSNs are unable to satisfy the performance requirements of CRSNs.

There exist many routing techniques for ad-hoc based cognitive radio networks (CRNs). Sampath et al. (2008) and Cheng et al. (2007) proposed an AODV based spectrum-aware routing protocol for CRNs. Sampth et al. (2008) aim to guarantee end-to-end performance by integrating flow-based approach with a link-based one. A routing metric is also derived based on the number of available channels. Cheng, et al. (2007) propose an on-demand routing protocol that selects suitable spectrums for each node along the path. According to spectrum availability, Pefkianakis et al. (2008) presented a routing solution called SAMER to provide long-term and short-term route, and Wang et al. (2009) proposed multi-path based routing protocol in order to improve connection stability. These efforts mainly focused on route and spectrum selection and considered expected performance including throughput, delay, and robustness. However, these methods cannot be directly applied to CRSNs because the energy efficiency is not considered as a design goal.

There is only limited work that focuses on routing issues in CRSNs. Parvin and Fujii (2011) proposed a spectrum-aware routing scheme with the goal of guaranteeing network QoS. More specially, they define a utility function which is used to evaluate the end-to-end delay, and a route with the maximum value is selected. Similar research is addressed by Shah et al. (2013). The authors design a distributed control algorithm to improve communication performance. Their primary goal is to guarantee the QoS requirements by optimizing an objective function, which is derived to minimize queueing delay (Shah et al., 2013). However, these routing schemes neglect the impact of unpredictable link failures on QoS.

In order to predict link failure and dynamically changing spectrum resources, a machine learning based routing solution is proposed in (Yu et al., 2010). Yu et al. (2010) employ the Bayesian learning, which is one of the supervised learning methods to estimate the total number of neighboring PUs. The estimated result is used to reflect the amount of available radio resources. Based on the estimation, the most reliable path that contains the largest number of available channels is selected. The major shortcoming of this routing scheme is that the requirement of the supervised learning based algorithm is ignored. In order to make an accurate estimation, the Bayesian learning needs a large amount of labeled data to train the learning algorithms. Unfortunately, labeled data is hard to be obtained in most of CRSNs' application scenarios because the network environment dynamically changes. Furthermore, their route configuring scheme neglects dynamic link failures in CRSNs. Therefore, their routing scheme cannot guarantee reliable end-to-end communications.

In contrast to the aforementioned related work, the proposed routing algorithm predicts the PUs' influence and evaluates link stability to guarantee reliable and stable communications.

3 PROPOSED SCHEME

The proposed routing algorithm consists of two parts: the *Context-Aware* module and *Optimal Path Configuration* module. The responsibility of the context-aware module is to learn the characteristics of the varying environment and predict the stability of routes. Based on that, the optimal path configuration module is used to estimate and select the most stable path to guarantee end-to-end communication reliability. The following subsections discuss these parts in detail.

3.1 Context-aware Module

In order to feasibly estimate the variations in the network environment, *semi-supervised learning*, which is a subcase of Machine Learning, is employed in our routing algorithm. Recently, Machine Learning has become one of the most efficient and practical solutions to solve several routing issues (Yu et al., 2010; Wang et al., 2006; Ahmed and Kanhere, 2010). Supervised and unsupervised learning methods are particular cases that perform learning tasks with labeled and unlabeled data, respectively. Wang et al. (2006), and Ahmed and Kanhere (2010) applied the supervised learning method to their research. However, their algorithms require a large amount of labeled data to train estimate functions. In most of CRSNs application scenarios, the labeled data are difficult and/or expensive to obtain. In contrast, the unsupervised learning method trains it's estimation function based on unlabeled data, which are readily available. However, the unsupervised method is more complex than its counterpart. Furthermore, the estimation accuracy cannot satisfy the communication requirements of CRSNs. These shortcomings can be alleviated by *semi-supervised learning*, which benefits from tactfully utilizing both labeled data and unlabeled data. The semi-supervised learning method can provide good learning accuracy even when there are only a few labeled data. Thus, it is more feasible than supervised or unsupervised method based one in CRSNs.

In order to learn the context information of CRSNs, the routing algorithm needs to gather some information as labeled data. To predict the link connectivity to sink nodes, each node needs to maintain the following context features:

- Neighbor node IDs $(D_1, D_2, ..., D_n)$, and sink node IDs $(S_1, S_2, ..., S_m)$.
- Current time slots $(t_1, t_2, ..., t_k)$ assuming that time of day T is divided into k slots.
- Currently available channel set $(Ch_1, Ch_2, ..., Ch_c)$ assuming a set of locally available channels.

In order to guarantee energy efficiency, the context information collection is performed in a passive manner. In the network initialization step, the sink node broadcasts HELLO message using a common control channel (CCC). In addition to the initialization step, the message is broadcast repeatedly with a period $(T+\varepsilon)/k$. During this period, ε is randomly selected within a time interval δ ($\delta \ll T/k$) to prevent congestion. The HELLO message is one of the control packets that contains sink nodes' IDs, locally available channel set, and a connectivity label Y (Yes) or N (No) field. A sink node sends the message after setting the connectivity label as Y. This message is identified by the sink node ID and the current time slot. A CRSD may receive many copies of the same message from one of the neighbors. In this case, the CRSD only forwards the first one to downstream neighbors.

When a node receives a HELLO message, it checks its label and the available channel set of the sender. If the label is Y and they have the common available channels, the node retains the label. Otherwise, the node changes the label to N. Before forwarding the message, the node stores the context information, which consists of sink ID, sender ID, the current time slot, and the label in local memory. Then, the localized context information is updated to the message. The node stops the forwarding until there are no downstream neighbors.

Through HELLO message forwarding, every CRSD can maintain the labeled context information and should know whether it is successfully connected to a sink node. For example, a context information $\langle S_1, t_2, D_5, \{Ch_2, Ch_3, Ch_4\}, Y\rangle$ indicates that the node can successfully communicate with sink S_1 at time slot t_2 using the relay node D_5, and the available channel set $\{Ch_2, Ch_3, Ch_4\}$.

The pseudo-code description of the proposed algorithm is given in Fig. 1. Let L_i denote the labeled context information and U_i denote all possible combination of the unlabeled context information in D_i. Two classifiers h_1 and h_2 are initially trained from L_i. Then, the classifiers are used to label $U_i \cdot u_i^k$, which indicate k^{th} unlabeled context information in D_i can be labeled if both of the classifiers agree on the labeling. When one of classifier disagree with the other one, the decision is made after comparing with a confidence threshold value τ. After labeling u_i^k, the node adds it to labeled context information and then obtains $L_i^{new} = L_i \cup u_i^k$. Then, the algorithm picks up u_i^{k+1} and repeats this procedure. This process is repeated until there are no more confident unlabeled data to be

```
input : L_i, U_i, h_1, h_2
output: L_i^new
for j ∈ {1,2} do
│ h_j ← Learn with L_i
end
repeat
│ for every u_i ∈ U_i do
│ │ if h_1(u_i)=h_2(u_i) then
│ │ │ L_i^new ← L_i ∪ {u_i, h_1(u_i)}
│ │ else if h_1(u_i) > τ then
│ │ │ L_i^new ← L_i ∪ {u_i, h_1(u_i)}
│ │ for j ∈ {1,2} do
│ │ │ h_j ← Learn with L_i^new
│ │ end
│ end
until none of h_1, h_2 change;
return L_i^new
```

Figure 1: Pseudo-code for the learning algorithm.

selected.

In this paper, we employ the *naive-Bayes* classifier and *decision tree learning* as the classifiers h_1 and h_2, respectively. The naive-Bayes classifier, which is based on the Bayes rule, is widely employed in posterior probability calculation with priori information. The naive-Bayes classifier is defined by the following equation:

$$h_1(u_i^k) = \underset{l \in \{Y,N\}}{\arg\max}\, P(l) \prod_j P(X_j|l), \qquad (1)$$

where X_j represents the context information value and l denotes the labels ($l \in \{Y, N\}$).

The decision tree learning makes decision by dividing the classification into a set of choices, and starting at the root of the tree and progressing down to the leaves (Marsland, 2009). Usually, the attribute that has the highest information gain for the label is selected as the root node and the parent nodes. The information gain can be calculated using the entropy of the attribute (Marsland, 2009).

The following example illustrates the proposed algorithm. Fig. 2 shows a CRSN where CRSDs occupy four licensed channels. We assume that a PU is moving through the CRSN during t_1 to t_2. With the HELLO message traverses the entire network, D_2 collects the localized context information as shown in Table 1, which shows both the labeled and unlabeled context information. After training the classifiers with the labeled data, they are used to classify the unlabeled data (the last four rows of the table). When the algorithm selects an unlabeled data $\langle S, t_1, D_4, Ch_4 \rangle$, which is obtained form the 5^{th} row in the table, the inference results is shown as follows:

$$\begin{aligned} P(Y) \prod_{j=4} P(X_{j=4}|Y) &= P(Y)P(S|Y)P(t_1|Y)P(D_1|Y)P(Ch_4|Y) \\ &= \tfrac{8}{10} \times 1 \times \tfrac{3}{8} \times \tfrac{5}{8} \times \tfrac{1}{8} \\ &= 0.023, \end{aligned} \qquad (2)$$

Table 1: Labeled and unlabeled data in the CRSD.

Destination	Time slot	Neighbor	Available channels	Label
S	t_1	D_4	Ch_1, Ch_2, Ch_3	Y
S	t_1	D_5	Ch_2, Ch_4	N
S	t_2	D_4	Ch_1, Ch_3	Y
S	t_2	D_5	Ch_2, Ch_3, Ch_4	Y
S	t_1	D_4	Ch_4	$?(N)$
S	t_2	D_4	Ch_2	$?(Y)$
S	t_2	D_4	Ch_4	$?(Y)$
S	t_2	D_5	Ch_1	$?(Y)$

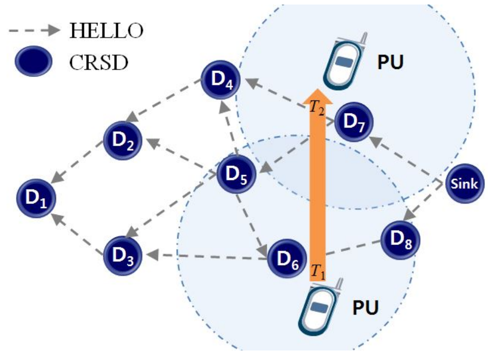

Figure 2: Context information gathering.

$$\begin{aligned} P(N) \prod_{j=4} P(X_{j=4}|N) &= P(N)P(S|N)P(t_1|N)P(D_1|N)P(Ch_4|N) \\ &= \tfrac{1}{4} \times 1 \times 1 \times \tfrac{0 + mp_{D_2}}{1+m} \times 1 \\ &= 0.042, \quad \textit{where} \ \ m = 10, \ \ p_{D_2} = \tfrac{1}{2}. \end{aligned} \qquad (3)$$

Eq. (2), and Eq. (3) indicate the packet delivery probability in Y and N cases, respectively. By substituting the results to Eq. 1, there is $P(N) \prod_{j=4} P(X_{j=4}|N) > P(Y) \prod_{j=4} P(X_{j=4}|Y)$, and then the algorithm obtains the inference result as:

$$h_1(u_2^1) = N. \qquad (4)$$

The other inference result can be obtained after constructing a decision tree[1] as shown in Fig. 3. Based on the decision tree, the inference decision becomes:

$$h_2(u_2^1) = N. \qquad (5)$$

Note that the intersection of the inference results is not empty; therefore, the unlabeled data is labeled as $h_1(u_2^1) \cap h_2(u_2^1) = N$. Then, the labeled data set L_2 is updated with the labeled data $u_2^1 = \langle S, t_1, D_4, Ch_4, N \rangle$, and a new set of labeled data is $L_2^{new} = L_2 \cup u_2^1$. The augmented labeled set can then be used to retrain the classifiers in an iterative manner until the termination condition is satisfied.

[1]We skip derivation process of the decision tree. Readers can refer to Chapter 6 in (Marsland, 2009).

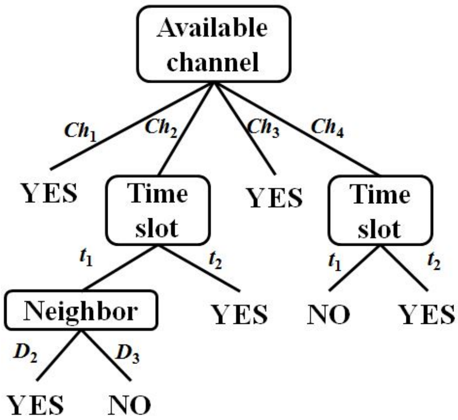

Figure 3: The decision tree.

Finally, the labeled context information is stored in local memory to further predict the stability of routes, and also is used to derive a novel routing metric for selecting optimal path as discussed in the following subsection.

3.2 Optimal Path Configuration

In order to evaluate the benefit of an intermediate node for path stability, a new routing metric called *Path Stable Metric* (*PSM*) is defined as follows:

$$PSM = \theta \sum_{j \in c} h_1(u_i^{new}) + (1-\theta)\frac{1}{n_c}\sum_{j \in c} h_2(u_i^{new}), \quad (6)$$

where c and n_c denote the available channel set and the number of channels in c, respectively, and θ $(0 \leq \theta \leq 1)$ is a weighted parameter for controlling the proportion of the two classifiers.

In this paper, we assume that only one channel can be occupied at a time. This indicates that any channel in the available channel set is independent of the others. Therefore, the routing metric must be derived by cumulating each channel's stability. The first term in the right side of the equation is obtained by $h_1(\cdot)$. The other one is obtained from $h_2(\cdot)$. If the label is Y, $h_2(\cdot) = 1$; otherwise, $h_2(\cdot) = 0$.

When a CRSD receives a routing request from the upper layer, it broadcasts route request (RREQ) packets (Perkins and Royer, 1999) on CCC. The message contains sink node ID and its currently available channel set. If the neighbors who receive this message have the common available channels with the sender, the nodes calculate local *PSMs*. Before forwarding the RREQ, intermediate nodes add its ID and *PSM*, and update available channel information with locally available channels. Finally, when the sink node receives the RREQ, a connectivity diagram is constructed based on contained intermediate nodes' ID and *PSM* information. Note that the sink node can receive multiple copies of the same RREQ packets from different neighbor nodes. In that case, the connectivity diagram should be updated only if a message contains a new *PSM* information. Every updated topology will trigger the semi-Dijkstra algorithm to find the path with the highest *PSM*. Then, the sink node sends a route reply (RREP) packet through the selected path. The RREP packet contains all the nodes' IDs along the path; therefore, the source node can send data packets to the sink node along the most stable path.

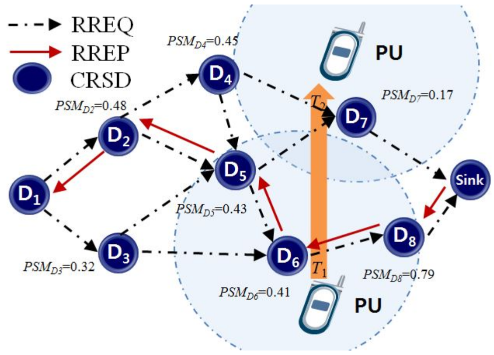

Figure 4: Optimal path selection.

In order to illustrate the optimal path configuration, consider again the context data learning example discussed in Subsection 3.1 and shown in Fig. 4. Suppose that node D_1 broadcasts a RREQ message, which consists of the current context information $\langle S, t_2, \{Ch_1, Ch_2\}\rangle$, at time slot t_2. When the neighbor node D_2 receives the message, it checks the intersection of available channel set. If they have a common set of available channels (In this case, their common available channel set is $\{Ch_1, Ch_2\} \cap \{Ch_1, Ch_2, Ch_3, Ch_4\} = \{Ch_1, Ch_2\}$, and $\theta = 0.7$), then the *PSM* is calculated as follows:

$$\begin{aligned}\theta \sum_{j \in c} h_1(u_1^{new}) &= \theta(P(Y)P(S|Y)P(t_2|Y)P(Ch_1|Y) \\ &\quad + P(Y)P(S|Y)P(t_2|Y)P(Ch_2|Y)) \\ &= 0.182,\end{aligned} \quad (7)$$

$$(1-\theta)\frac{1}{n_c}\sum_{j \in c} h_2(u_1^{new}) = (1-0.7)\frac{1}{2}(1+1) = 0.3. \quad (8)$$

By substituting the results to Eq. (9), *PSM* for D_2 is given by

$$PSM_{D_2} = \theta \sum_{j \in c} h_1(u_1^{new}) + (1-\theta)\frac{1}{n_c}\sum_{j \in c} h_2(u_1^{new}) = 0.482. \quad (9)$$

*PSM*s for the other nodes in Fig. 4 are derived in the same manner.

After receiving the RREQ message, the sink node can construct a network topology as shown in Fig.

4. Using the semi-Dijkstra's algorithm, the sink node can find the most reliable and stable path and then reply to D_1 with a RREP.

4 PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed scheme through simulation. We randomly deploy 100 CRSDs in a simulated area 100×50 m^2. The packet size is set to 100 bytes, and one CCC and four licensed channels are available in the simulation environment. We assume that the transmission range of CRSD is 5 m and the interference rage is 10 m. The activity of PUs is a Poisson process with arrival rate λ. We also assume that CRSD stops transmission immediately if it suffers from PUs' interference. Since the naive-Bayes classifier has been shown to be more stable and provide better performance than decision tree based methods (Marsland, 2009), the proposed routing algorithm uses θ=0.7 indicating that more trust is given to the naive-Bayes classifier.

The proposed routing algorithm is compared with SAMER (Pefkianakis et al., 2008), which uses the minimum hop count as the route selection metric. Fig. 5 compares the communication reliability of the two routing algorithms. For each simulation run, the arrival rate λ is randomly selected in the interval of $0.15-0.5$. We can observe from Fig. 5 that the packet delivery ratio is not stable when the network employs minimum hop-count as the routing metric. This is because the performance of SAMER is significantly affected by how often the minimum hop-count path is interrupted by PUs. As the simulation proceeds, the average packet delivery ratio of SAMER converges to 72% which cannot satisfy most QoS requirements. In comparison, the proposed semi-supervised based routing algorithm shows more stable packet delivery ratio and the average performance converged to 92%. More specifically, the proposed routing protocol provides 28% performance gain over SAMER. There are two main reasons for this. First, the proposed Context Aware module can intelligently estimate the scalability of the path. Second, the most stable path, which has the lowest possibility of interruption, can be effectively established based on the proposed routing metric. Furthermore, the proposed method can efficiently avoid PUs' influence; therefore, extra energy consumption caused by frequent retransmissions can also be efficiently avoided.

Fig. 6 shows that the proposed method clearly outperforms SAMER in terms of throughput. As the arrival rate of PUs increases, the throughput for SAMER drastically decreases. This is because the active behavior of PUs has a significant impact on the system throughput. Although SAMER has benefits in terms of minimum hop routing and low transmission delay, it cannot provide satisfactory throughput due to the influence of PUs. The proposed routing algorithm shows more stable performance than SAMER, even when the arrival rate of PUs is increased. This is due to the fact that the proposed routing algorithm reduces the probability of route failure by learning and estimating the activity and channel utilization of PUs.

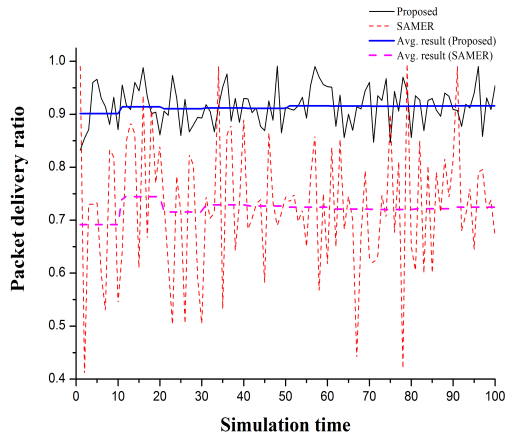

Figure 5: Packet delivery ratio.

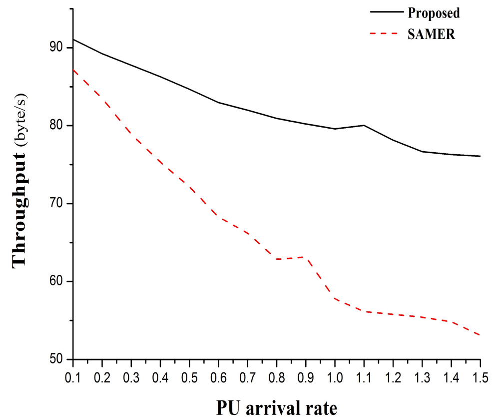

Figure 6: The network throughput

5 CONCLUSIONS

This paper presented a semi-supervised learning based routing algorithm that predicts and minimizes the influence of PUs. The proposed routing algorithm consists of Context-Aware and Optimal Path Configuration modules, and uses a feasible semi-supervised learning algorithm to perceive variation of the network connectivity. In addition, a novel routing metric is derived to estimate end-to-end connection stability. The optimal path, which is the most stable in the network topology, is selected using the semi-Dijkstra algorithm. Finally, our simulation study shows that the proposed method effectively provides reliable and stable paths, and improves system throughput. We are currently in the progress of designing an energy-efficient cross-layer routing algorithm, which combines spectrum selection with route configuration by applying a feasible machine learning method.

ACKNOWLEDGEMENTS

This work was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2013-(H0301-13-2001)) and by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2011-0015744).

REFERENCES

Ahmed, S. and Kanhere, S. (2010). A bayesian routing framework for delay tolerant networks. In *Proc. of WCNC'10*.

Akan, O. B., Karli, O. B., and Ergul, O. (2009). Cognitive radio sensor networks. *IEEE Network Magazine*, 23(4).

Ali, A., Iqbal, M., Baig, A., and Wang, X. (2011). Routing techniques in cognitive radio networks: A survey. *International Journal of Wireless & Mobile Networks*, 3(3).

Cesana, M., Cuomo, F., and Ekici, E. (2011). Routing in cognitive radio network: Challenges and solutions. *International Journal on Ad Hoc Networks*, 9(3).

Cheng, G., Liu, W., and Cheng, W. (2007). Spectrum aware on-demand routing in cognitive radio networks. In *Proc. of DySPAN'07*.

Marsland, S. (2009). *Machine Learning: An Algorithmic Perspective*. A Chapman & Hall.

Pefkianakis, I., Wong, S. H. Y., and Lu, S. (2008). Samer: Spectrum aware mesh routing in cognitive radio networks. In *Proc. of DySPAN'08*.

Perkins, C. E. and Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *Proc. of WMCSA'99*.

Sampath, A., Yang, L., Cao, L., and Zhao, B. Y. (2008). High throughput spectrum-aware routing for cognitive radio networks. In *Proc. of CROWNCOM'08*.

Shah, G., Gungor, V., and Akan, O. (2013). Routing in cognitive radio network: Challenges and solutions. *IEEE Tran. on Industrial Informatics*.

Wang, X., Kwon, T., and Choi, Y. (2009). A multipath routing and spectrum access framework for cognitive radio systems in multi-radio networks. In *Proc. of ACM Workshop on Cognitive Radio Networks*.

Wang, Y., Martonosi, M., and Peh, L. S. (2006). Supervised learning in sensor networks: new approaches with routing, reliability optimizations. In *Proc. of SECON'06*.

Yu, R., Zhang, Y., Yao, W. Q., Song, L. Y., and Xie, S. L. (2010). Spectrum-aqare routing for reliable end-to-end communications in cognitive sensor networks. In *Proc. of IEEE Globecom'10*.