2016 IEEE 36th International Conference on Distributed Computing Systems

CACA: Link-based Channel Allocation Exploiting Capture Effect for Channel Reuse in Wireless Sensor Networks

Junghyun Jun[1], Solchan Yeon[2], Titir Kundu[3], Dharma P Agrawal[3], and Jaehoon (Paul) Jeong[4]

[1]Indian Institute of Technology Ropar, India
[2]Kookmin University, Republic of Korea
[3]Department of EECS, University of Cincinnati, Ohio State, USA
[4]Department of Interaction Science, Sungkyunkwan University, Republic of Korea
peterjun@iitrpr.ac.in, green1343@kookmin.ac.kr, kundutr@mail.uc.edu, dpa@cs.uc.edu, pauljeong@skku.edu

***Abstract*—In this paper, we exploit the capture-effect for channel allocation. We experimentally show the characteristics of capture-effect across different channels, over time, and in different network densities. Then, we introduce CACA, an effective channel assignment protocol for wireless sensor networks. Traditional channel assignment protocols utilize all available channels to minimize interferences between any adjacent links. However, their performances are often not much better than the case of using single channel only. This is mainly due to an assumption that all channels are independent and quality of all channels are similar. However, this is a false assumption. In fact, there are only a few channels that show very good quality at any given time. The CACA avoids this problem by utilizing a few good channels and reuse these channels. When the channels are reused it relies on the *capture-effect* to ensure at least one of the contending nodes to transmit successfully. Maximizing this capture probability is a main objective of the CACA whenever the channels need to be reused. We evaluate the CACA on a 140-node wireless sensor network testbed and compare its performance with another benchmark protocol. Our results indicate that the CACA can improve the packet reception ratio of every link. As a result of this improvement, end-to-end throughput increases upto 100% in the case of a wireless sensor network with bursty traffic.**

***Index Terms*—channel diversity; capture-effect; wireless sensor network;**

I. INTRODUCTION

A channel assignment protocol selects a channel for each link in the network. Applying different channels to any neighboring nodes can eliminate their interferences, unless their co-channel interference are strong. Therefore, this is a common operation for a wireless sensor network (WSN) with bursty traffic. Especially, for the low-duty cycle networks, any packet loss can significantly increase the network latency and reduce its throughput. Therefore, minimizing the interference by exploring the channel diversity has become an important requirement for low-duty cycle WSNs.

In WSNs, IEEE 802.15.4 specifies 16 channels. But, the qualities of some channels are poor due to noisy environment such as coexisting WLAN. Many WSNs first blacklist those poor quality channels before the channel assignment [1]. Therefore, there may not exist enough channels to completely remove all interference. Many traditional channel allocation protocols do not consider this situation [2]–[5]. There are some channel allocation protocols to handle the case of insufficient number of channels. However, they often ignore varying channel qualities [6]–[8]. They simply assumes all channels are similar and independent.

An experiment result from [9] shows that the channel quality of a link varies significantly across different channels due to different external interferences. Therefore, if a network adds a new channel to the existing channel allocation, it may cause many poor quality channels for the links. Consequently, the network may gain very little from channel diversity or it could end up degrading the network performance that is achievable by the network utilizing one good quality channel. Based on this observation, the work in [9] proposes an ILTP protocol which effectively utilizes intermediate quality links by switching between four good quality channels.

In WSNs, unlike the other wireless networks, each sensor device can not simultaneously operate on different frequencies. Therefore, utilizing many channels requires a frequent channel switching. Although the switching time of transceivers has been continuously reduced, a modern transceivers like CC2500 still takes 90 microseconds to switch between two channels [10]. Therefore, this additive overhead can be significantly reduced by limiting the available channels to low number in a large-scale WSN.

Besides all these benefits of limiting the available channels, there is one major drawback. In dense WSNs, some links must operate with the same channels since many nodes have more outgoing links than the available number of channels. Therefore, limiting the available channels increases potential packet collisions. The existing RTS/CTS protocols are often utilized to avoid such collisions. However, the packet size in WSNs is very small and comparable to the size of RTS and CTS packets. Since its associated overhead is high, the RTS/CTS protocols are not suitable for WSNs. The MMSN protocol adopted channel listening in order to reduce concur-

1063-6927/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDCS.2016.65

IEEE computer society

rent transmissions and avoid the direct collisions. However, it increases packet delay and still suffers from the problem of hidden terminal. More recent work in [11] proposed a channel allocation algorithm minimizing the maximum number of conflicting links.

Different from all existing channel allocation schemes, we propose a capture-effect aware channel allocation (CACA). Instead of minimizing the conflicting links, CACA reduces a total number of packet losses due to collisions by exploiting the capture-effect. The capture-effect is a physical phenomenon which allows a node to successfully receive one of the messages under the event of collision. It carefully selects a set of links to share a common channel such that the benefit of this capture-effect can be maximized. The basic idea behind the CACA is to first receive both packets from a concurrent transmission by channel diversity. If the reception of both packets is not possible, receiving one of them is still better than losing both packets. Our hypothesis is that a channel reusing policy following this approach will help most links in the network to maintain a good throughput.

It is also well-known that the capture-effect is actually ineffective on the links having more than two other interfering links. Therefore, in order to take a full advantage of this capture-effect, it is recommended for a link to avoid a highly conflicting channels. We include an indirect channel balancing mechanism in our channel assignment process such that the channels are evenly distributed among the links.

We evaluate CACA by running it on a 140-node wireless testbed called Indriya that uses TelosB devices. Our extensive experiment shows that exploiting the capture-effect in the channel assignment can significantly improve the individual link success ratio and increase an overall network throughput. Our evaluation results show that CACA can improve an overall packet reception at every link in the network by using even three channels. In case of a WSN with good link quality and good capture-effect, CACA can improve total end-to-end throughput up to 100% when the traffic is bursty and 170% if the channel contention ratio is reduced by 50%. To the best of our knowledge, the capture-effect have never been exploited for the channel allocation before.

II. RELATED WORKS

Utilizing capture effect: Capture-effect has been well studied in the domain of WSN. Lu et al. [12] demonstrated in a 48-node testbed that flashing flooding can reduce the latency of the overall network by 80% when capture effect is exploited. Capture-effect has also been studied to detect collision and recover data [13]. Son et al. [14] showed in their work the effect of concurrent transmission and capture to study the SINR value with regards to network density in WSN.

Capture-effect has been exploited in the domain of concurrently transmitting the copies of a same packet under strict time synchronization such as a flooding. This type of transmissions leads to a phenomenon called a constructive interference. This phenomenon was first exploited to improve WiFi network performance in the work of [15]. Along with improvement of the download throughput in the infrastructure network, it also contributed to the performance enhancement of ad hoc mesh networks for opportunistic routing [16]. Glossy [17] demonstrates that synchronous transmission leads to co-operative interference and this can improve the performance even in the absence of capture. But, in their work, they also mention that co-operative interference can happen if the time displacement is no more than 0.5 microsecond in a standard CC2420 sensor [18]. Another protocol to exploit constructive interference is Splash [19] which disseminates bulk amount of data across all nodes in a WSN.

Capture-effect under synchronous transmission of different packets is demonstrated in Chaos [20] such that the average delay in all-to-all communication can be significantly reduced. In Chaos, the capture-effect is considered for synchronous transmission. It transmits different packets in an all-to-all communication by exploiting the capture-effect.

For the first time, we studied capture-effect for an efficient channel allocation. The channel assignment is an imperative issue in many wireless networks and many promising approaches have been proposed. But none of them directly exploited the capture-effect. We consider some of the well-known channel assignment problems.

Channel assignment problem: Utilizing multiple channels is a promising approach to improve the performance of wireless networks. Assigning different channel to each wireless link is the simplest technique that could increase the network capacity. The key idea of this technique is to minimizing the channel interference between spatially close links. Many algorithms and approaches are proposed to solve this channel assignment issue.

In a WSN, there are many distinctive approaches to efficiently assign multiple channels. One approach is a greedy, tree-based, multi-channel protocol called Tree-Based Multi-Channel Protocol (TMCP) [21]. In this approach, the network is divided into multiple sub-trees and intra-tree interference is reduced by assigning different channels to each subtree from the top of the tree to the bottom. Also, a simple greedy approach has been proposed in [21] in order to increase the parallel transmissions.

Another approach is a game theory based solution for channel assignment [22]. It performs channel assignment by considering traffic volumes carried per node, which exploits routing and topology information to reduce contentions. This approach resulted in the improvement of multichannel MAC performance. A most recent approach addresses a case of insufficient number of available channels [11]. It focuses on minimizing the maximum interference per link by allocating the available channels. It first creates a receiver-based conflict graph and locally allocates channels based on the degree of conflicting links.

There are other related works which attempt to jointly optimize channel assignment while considering residual energy of nodes. Li [23] proposed residual energy-based channel assignment to improve performance in WSNs. An R-co-efficient has been developed with the knowledge of current residual energy

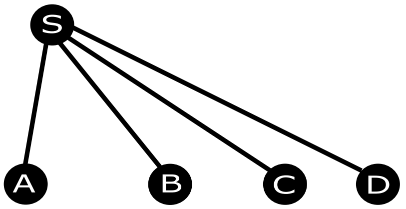


Fig. 1. An example of 5 nodes WSN with a single channel

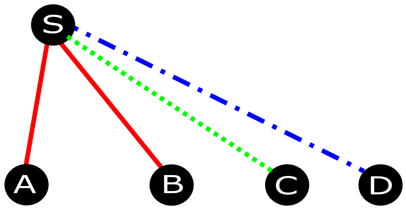


Fig. 2. An example of 5 nodes WSN with channel diversity

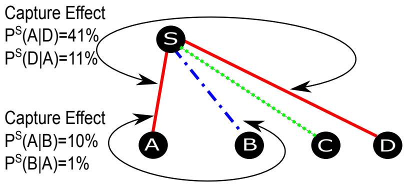


Fig. 3. An example of 5 nodes WSN with capture effect

and expected future energy consumption, considering the channel condition. The R-efficient-based channel assignment protocol out performed random pairing and a greedy channel search.

None of the above work considered the capture effect while optimizing the performance by channel assignment. Most of them also assume a large number of channels is available. Typically, WSN devices cannot handle multiple channels simultaneously due to their small form factors. Our work considers an optimal channel allocation problem for the WSN with an insufficient number of available channels. Yet we achieve a good performance by effectively exploiting the capture-effect. So in our best of knowledge, our work is a very first channel reuse approach to optimize the utilization of capture-effect in WSNs.

III. Problem Statement

In a WSN, several protocols explore channel diversity so as to improve the throughput and reduce the number of retransmissions. Multiple channels allow multiple links to concurrently transmit their packets without experiencing any interference. In general, their objective is minimizing such interference among their adjacent nodes. Assigning the channel to a node to receive a packet from any senders is commonly known as receiver-based channel allocation. Whereas, in link-based channel allocation, each link is assigned with a channel and uses it for any transmission along that link. Both problems is often reduced to a graph coloring problem, where a color represents a specific channel in a WSN. But, our problem mainly focuses on link-based channel allocation.

Formally, a graph $G = (V, E)$ represents the network connectivity, where a vertex $v \in V$ presents a sensor node and an edge $e \in E$ represents a connectivity between a pair of nodes. The number of available channels (colors) is fixed to K (for example, $K = 16$ for a sensor node with CC2420 RF transceiver). The problem in coloring the edges is to minimize the number of adjacent edges with the same color. For example shown in Figure 1, there are 5 nodes and all the nodes opportunistically forward data to a node S. If all 5 links use a single frequency channel, packet loss ratio would be high due to collisions. Ideally, one can eliminate collision by assigning a unique channel to each edge. However, often we cannot expect to utilize too many channels in a WSN due to the low availability of good channels and high overhead.

Suppose we only have three available channels. Let us represent each channel with red, green, and blue, respectively. Figure 2 shows an example of 3-channel assignment. Utilizing three good channels would certainly improve the throughput as compared to a single channel case. Still, we could expect some packet collisions between links (A,S) and link (B,S) which share a common channel. There are six possible combinations of three channel assignment for the network in Figure 2. This raises an important question.

Which one would be the best channel assignment in the case where an insufficient number of channels are available?

Perhaps all the combinations are the same if their link qualities are the same. However, we know that not every concurrent transmission results in a collision. This is mainly due to a physical phenomenon known as a capture-effect. The capture-effect allows to recover one packet from a collision. That is, it recovers the packet with the highest RSS or packet arriving earlier than the other packets. The capture effect requires that the packet to be captured must have RSS significantly higher than that of the other packets (in the collision) and this RSS difference may vary depending on the modulation technique. Therefore, without considering this capture-effect, the resulting channel allocation could be a suboptimal channel combination.

In order to apply this capture-effect in channel allocation, let us now formally define a pair-wise capture-effect probability as follows:

Definition 1: A capture probability is defined as $P^S(A|B)$, a conditional probability of a node S successfully receiving a packet transmitted from node A due to a capture given the condition that both nodes A and B transmit and their packets collide at node S.

For example, in Figure 3, the capture probabilities are $P^S(A|B) = 0.1$, $P^S(B|A) = 0.01$, $P^S(A|D) = 0.41$, and $P^S(D|A) = 0.11$. This means, if the same channel is assigned to nodes A and D, their concurrent transmissions would normally result in a collision. However, node S could still receive one of them successfully with probability of $P^S(A|D) + P^S(D|A) = 0.52$ due to a capture-effect. Similarly, if the same channel is assigned to nodes A and B, node S could still receive one of them with probability of $P^S(A|B) + P^S(B|A) = 0.11$. If there are only three available channels, assigning the same channel to both links (A,S) and (D,S), is certainly a better choice than assigning the same channel to links (A,S) and (B,S). Additionally, this will reduce the possibility of two retransmitted packets to collide again.

This example clearly illustrates the importance of consider-

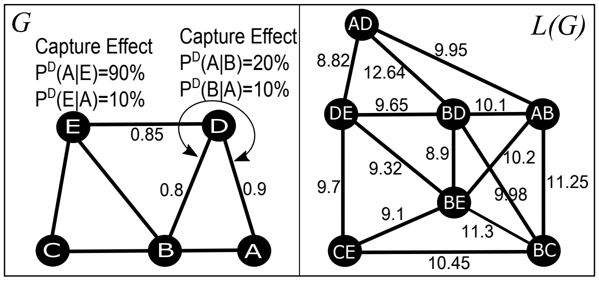


Fig. 4. Converting Vertex-to-Edge Dual Graph

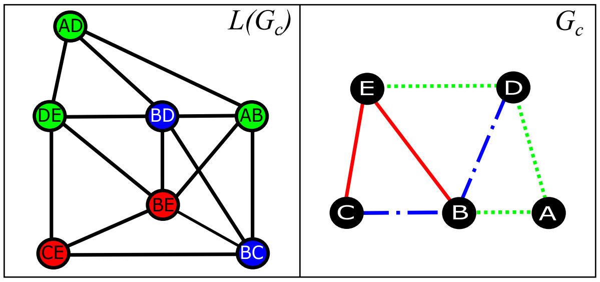


Fig. 5. 3-Channel Assignment on $L(G)$

ing capture effect in the channel assignment. Next, we propose an approach to effectively exploit capture-effect in the channel assignment of a WSN.

IV. CAPTURE-EFFECT AWARE CHANNEL ALLOCATION

In multi-frequency MAC protocols, a frequency channel is assigned to each link for data transmissions. Often these protocols have two aspects: channel assignment and media access. Most of the time, channel assignment is done once in the beginning of system deployment by the base station (centralized) or distributed by each node sharing their channel selection with its neighbor nodes.

One of the main advantages of CACA is its simplicity. It can be a standalone channel assignment scheme or a plug-in for other existing protocols.

When it used as a standalone channel assignment, a base station collects link qualities and capture-probabilities from every node before the channel allocation process. Many wireless sensor network applications rely on data collection protocols such as CTP [24] which updates and maintains link state information. Therefore, it makes more sense for channel allocation to take place at the base station along with collection protocol providing capture-probabilities.

When it is used as plug-in, it basically shuffles current channels to take advantage of capture-effect without changing the original channel distribution. In this case, each receiver can apply CACA locally if any one pair of its links is experiencing a significant change in its capture-effect.

Before applying CACA, the network must be able to detect the events for capture-effect. This can be done as a part of collision detection mechanism [13], [25]. The main CACA consists of three phases. The first phase converts an original network graph to a capture graph. In the second phase, we divide all nodes into K groups ($K = 3$ if there are three available channels) such that it maximizes the benefit of total capture effects. A node in the capture graph represents an edge in the original network graph and an edge in the capture graph represents a common node between two adjacent edges in the original network graph. The last phase basically assigns a channel to each link in the network by converting the capture graph back to original network graph after the second phase.

A. Capture-effect Event Detection

Accurately estimating the capture-effect probability is a crucial requirement of CACA. However, unlike measuring link quality, detecting occurrence of capture-effect is not a straight forward procedure. One method proposed in [25] basically detects the capture-effect event by overhearing ACKs and sharing the transmission timestamps. For example, a capture-effect is detected if one overhears an ACK and learns later that their transmissions are colliding. This method incurs an additional overhead of overhearing others' ACKs and exchange of beacon messages. Another method proposed in [13] directly detects the capture-effect. It basically detects the capture-effect from continuously searching for the preamble part of the packet while receiving the signal. While this is an overhead-free method, its capture-effect detection ratio is only around 50%. It can only increase the detection ratio close to 100% by adding the preamble part at the end of every packet. This requires a modification to the original packet structure.

For evaluating the CACA, we adopted the first method proposed in [25]. Although it adds an additional overhead, its detection ratio is almost 100% and our evaluation collects the capture effect once in the beginning of the experiment. The overhead of detecting capture-effect using this method is $4\binom{|E|}{2}$, since the transmissions of 3 beacons and 1 ACK are required for every pair of links.

For many WSNs, energy efficiency is their primary concern. Therefore, when applying the CACA to any real applications, it would be preferable to adopt the second method proposed in [13] with modification to the original packet structure. Its overhead is $2\binom{|E|}{2}$ since it requires 2 beacon transmissions for every pair of links.

B. Constructing Capture Graph

By Definition 1, capture-effect is an event involving a pair of edges. Therefore, our focus is on the edges rather than the vertices of a network graph. We first construct an edge graph (a line graph) $L(G)$ of an original network graph G. Each node in $L(G)$ corresponds to an edge in G and an edge in $L(G)$ corresponds to a common node between two adjacent edges. Figure 4 shows an example of constructing the edge graphs from the simple network graph with 5 nodes. The vertex (AD) of $L(G)$ in Figure 4 is corresponding to the edge (A, D) in G.

The edge weight in $L(G)$ is computed by comparing an individual pair's link quality and its associated capture effect. For example, a pair of edges (A, D) and (B, D) of G in Figure 4, have the link qualities of $P(A, D) = 0.9$ and $P(B, D) = 0.8$, respectively. The capture probabilities of that pair of edges are $P^D(A|B) = 0.2$ and $P^D(B|A) = 0.1$. The edge weight of (AD, BD) in $L(G)$ is computed as $(1/P^D(A|B) - 1/P(A, D)) + (1/P^D(B|A) - 1/P(B, D)) = (1/0.2 - 1/0.9) + (1/0.1 - 1/0.8) = 12.64$.

The edge weights in $L(G)$ represent the trade-off between a single channel and multi-channels. When two edges $(A, D), (B, D) \in G$ operate with different channels, node D would expect to receive a packet within $1/0.9 + 1/0.8 = 2.36$ concurrent transmissions from node A or node B. However, when these two edges utilize the same single channel, expected number of concurrent transmission increases to $1/0.2 + 1/0.1 = 15$. Here, the trade-off between a single channel and multiple channels is the additional transmissions of $15 - 2.36 = 12.64$. All edge weights of $L(G)$ are computed in a similar manner.

The capture-effect can occur for higher than two concurrent transmissions. In fact, 3-packet-capture probability can be computed from 2-packet-capture [26]. However, we did not directly consider 3-packets-capture scenario here since three packet collision probabilities are smaller than two packet collisions and its capture probability is much smaller than the case of two packets. But, we attempt to minimize such occurrences by balancing the channel distribution in the channel assignment phase.

C. K-Channel Assignment as Max K-Way Max Cut

When available channels are not enough and potential conflicts exist, it is better to reuse the channel on a pair of edges with the smaller trade-off. This is demonstrated from the example in Section III using Figure 3. Following this intuition, we design an Algorithm 1. It is a bottom-up greedy channel assignment scheme. After constructing a capture graph $L(G) = (L(V), L(E))$, where $L(V)$ is a set of vertices and $L(E)$ is a set of edges. Each edge $e \in L(E)$ is associated with its edge weight $w(e)$.

Overall, Algorithm 1 attempts to minimize the total edge weights (total trade-off). It first exhaustively assign K channels to pairs of links that have no benefit of utilizing their capture-effect. It achieves this by first sorting the edges based on their edge weights. Then, it starts assigning different channels to the pairs of vertices $(u, v) \in \overline{L(E)}$ with the high edge weights. Consequently, a pair of vertices with high edge weight generally shares the same channel, $u \in C_i$ and $v \in C_i$. This is a desirable outcome since the higher edge weight in capture graph means that the packet loss ratio of two links is high due to their collisions. But, their individual link qualities are still good. For example, in Figure 5, the 3 channels such as red, blue, and green, are assigned to every $v \in L(V)$ by following the algorithm 1. The edge weight of $(AD, BD) \in L(E)$ is 12.64. This is a largest edge weight in $L(G)$. So, different color channels are assigned to nodes (AD) and (BD). Consequently, both nodes (AD) and (DE) end up with a green channel. However, this is not bad since link $(A, D) \in G$ gains benefit from the capture-effect between (A, D) and (E, D) with $P^D(A|E) = 90\%$ compared to $P^D(A|B) = 20\%$.

Algorithm 1 K-channel assignment

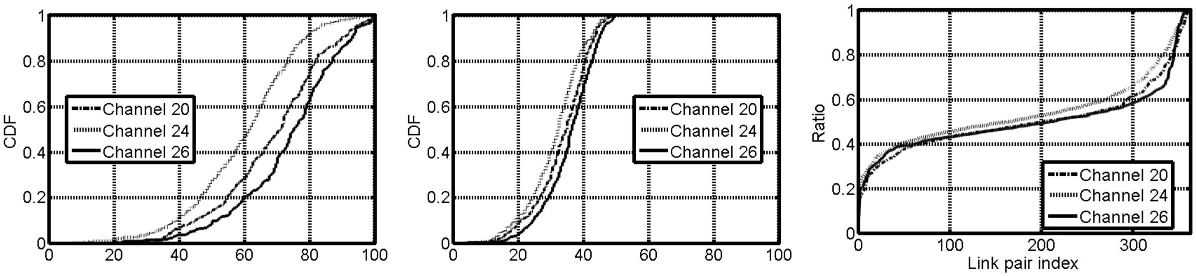
```
1: procedure KChannelAssign
2:     $C_i = \emptyset, i = 1 \ldots K$
3:     $\overline{L(E)} = Sort(L(E))$ descending order of $w(e)$
4: repeat
5:     $(u, v) \leftarrow e \in \overline{L(E)}$
6:     $\overline{C} = Sort\{C_1, \ldots, C_K\}$ ascending order of $|C_i|$
7:     $C_1 \in \overline{C}$
8:     $C_2 \in \overline{C}$
9:     if both u and v have no channel then
10:        $C_1 \leftarrow u$
11:        $C_2 \leftarrow v$
12:    if u has no channel but $v \in C_k$ then
13:        $C_1 \leftarrow u,$ where $C_1 \neq C_k$
14:    if $u \in C_k$ but v has no channel then
15:        $C_1 \leftarrow v,$ where $C_1 \neq C_k$
16: until all $v \in L(V)$ has a channel.
```

Notice from steps 6-8 in Algorithm 1 that when the channel need to be reused, it assigns one of the least utilized channels. This is an attempt to balance the channel distribution and is important since many concurrent transmissions result in packet loss, despite the capture effect [27].

D. Analysis of CACA

The main part of CACA consists of constructing a capture graph and K-channel assignment. When at most 2-packet-capture is considered in the constructing capture graph, its time complexity is $O(\binom{|E|}{2} - |E|)$. This is directly proportional to the number of edges $|L(E)|$ in $L(G)$. During the conversion from G to $L(G)$, every pair of conflicting edges in E forms a new edge in the $L(G)$. The number of pairs of edges in G is $\binom{|E|}{2}$. Therefore, the number of pairs of conflicting edges is $\binom{|E|}{2} - |E|$ since every edge $e \in E$ is counted twice. Next, K-channel assignment algorithm first sorts edges $L(E)$ in $O(|L(E)|log|L(E)|)$ time. Steps 5 to 15 is executed $|L(E)|$ times. At each execution, step 6 sorts available channels in $O(KlogK)$ time. The remaining steps take constant time for comparing and updating. Totally, step 6 is computed for $O(|L(E)|KlogK)$ times. Since $|L(E)| = O(\binom{|E|}{2} - |E|)$, the time complexity of CACA is $O(|L(E)|max\{log|L(E)|, KlogK\})$.

When the number of available channels K is large, prior to CACA, one should first determine the minimum number of channels required for conflict-free channel allocation. This can be done by employing a well-known greedy vertex-coloring algorithms like WelshPowell algorithm [28] on $L(G)$. The WelshPowell algorithm first requires the sorting of edges based on their degrees. Therefore, its time complexity is



(a) Link quality (Packet reception rate) (b) Capture-effect probabilities (%) (c) Capture-effect and link quality ratio

Fig. 6. Link quality distribution and Capture-effect distribution in Indriya testbed at different channels.

$O(|L(E)|log|L(E)|)$. By the same analogy used in [28], we can proved that it is always feasible to find a conflict-free channel allocation for $L(G)$ with at most D channels, where D is the max degree of $L(G)$. Since $D \leq |L(E)|$, the time complexity of the CACA would always be $O(|L(E)|log|L(E)|)$ in practice.

V. EVALUATION

In this section, we apply CACA for 3-channel assignment and experimentally evaluate its performance gain by comparing its to a single channel network and Eavesdropping. The Eavesdropping is a multi-channel assignment scheme proposed in [29]. Its low overhead design is suitable for WSNs. It is also a well-known benchmark protocol [22], [30].

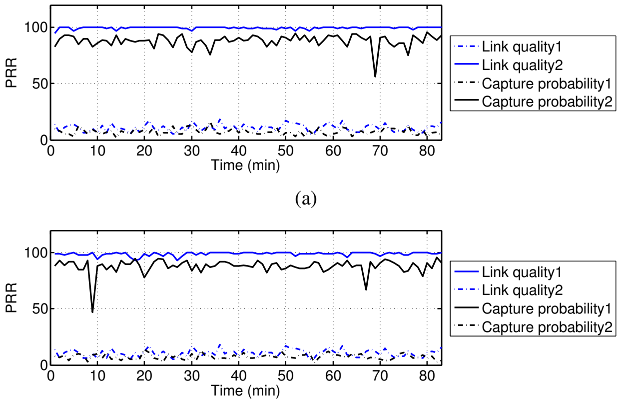
A. System Overview and Insight

We conduct our experiment in 140-node Indriya testbed by varying the transmission power of all nodes from level 6 to level 16. This creates 6 different random topologies with different node densities. Table I presents average node densities and the maximum node degree of Indriya testbed at 6 different transmission powers. Table I also shows the minimum number of channels required for conflict-free network at 6 different transmission powers. It was computed by link-based channel assignment algorithm proposed by [11].

For each topology, we first form a capture table containing the link quality (packet reception rate) and capture probability of every pair of links in the network. Figure 8 shows the temporal variation of link quality and capture-effect of two different link pairs in Indriya testbed. This result indicates that when the link quality level stayed steady, their corresponding capture-effect also does not change. Therefore, it is not necessary to regularly update the capture table. One can update the table whenever there is some noticeable change in the link quality.

TABLE I
NODE DENSITIES W.R.T. TRANSMISSION POWERS

Transmission power	6	8	10	12	14	16
Average number of neighbors	5	7	8	9	10	11
Max degree	11	15	17	17	18	22
Min number channels	5	5	7	10	10	9



(a)

(b)

Fig. 8. Temporal capture probabilities with associated link qualities

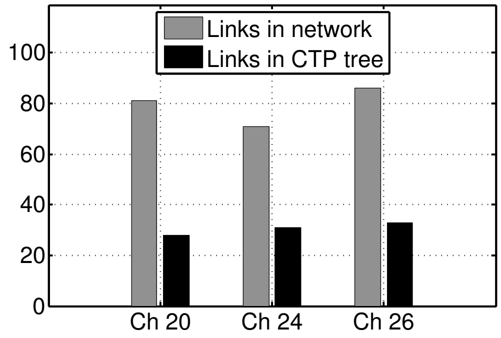
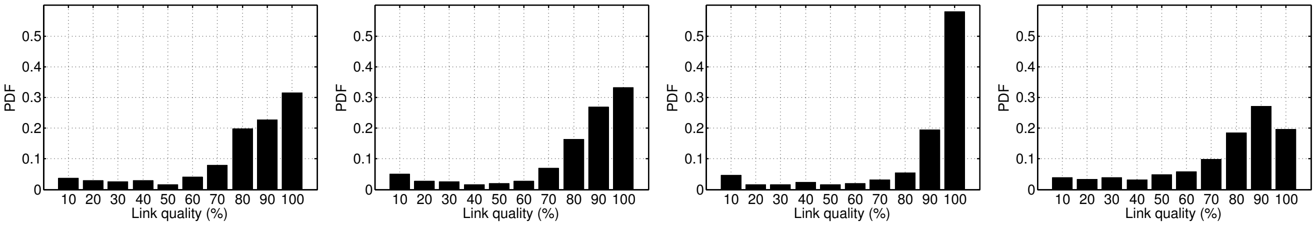
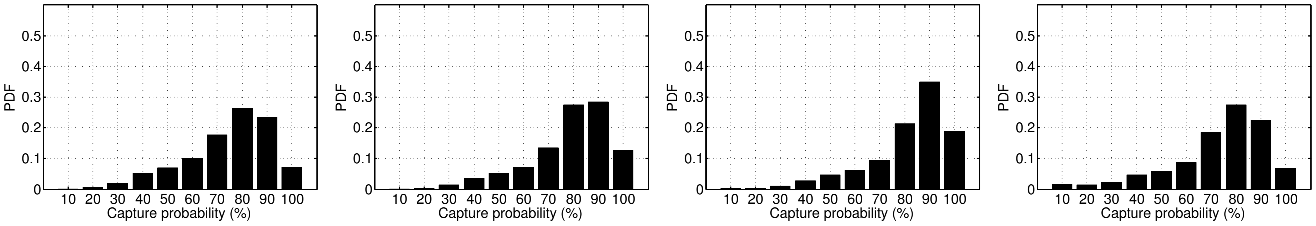


Fig. 9. Channel distribution after allocating channels by Eavesdropping

The channels 20, 24, and 26 of CC2420 transceiver are utilized in our channel assignment evaluation. These channels provide relatively a good set of links compared with other channels. Figure 6(a) shows the link quality distribution at different channels. The average link quality at Channel 26 is around 60% and it is the best among the three channels. The same behaviour has been observed by Manjunath in [9]



(a) TxPw = 6 (b) TxPw = 8 (c) TxPw = 10 (d) TxPw = 12



(e) TxPw = 6 (f) TxPw = 8 (g) TxPw = 10 (h) TxPw = 12

Fig. 7. Link quality distribution and Capture-effect distribution in Indriya testbed with different transmission powers (TxPw).

from the Indriya testbed. Figure 6(b) shows the capture-effect probability distributions. They do not vary much between these three channels. We also took a ratio between link quality and capture-effect probability of a pair of links (e.g., $P(A,S)/P^S(A|B)$) and compared their ratios across different channels. Interestingly, we can observe from Figure 6(c) that their ratios remain similar across all the three channels.

Figure 7(a), 7(b), and 7(c) show that the overall link qualities in the Indriya testbed is above 80% and are improved with increasing transmission power. Similarly, Figure 7(e), 7(f), and 7(g) show that the pairs of links experiencing good capture-effect increase according to the improvement of link qualities. Notice from 7(h) that the overall link qualities and the capture-effect decrease when the transmission power is set to 12. This is mainly due to the fact that increasing transmission power added new poor quality links to the network. This also reduces the overall quality of the capture-effect since capture-effect probability between any pairs of link is upper-bounded by their link qualities.

For packet forwarding, CTP protocol is used to construct a initial multi-channel routing tree [24]. Every node transmits one packet at every 10 msec and each node generates 100 packets destined to a root node. For channel access, we used ALOHA MAC protocol. When a node has a packet it forwards the packet to its parent node and then waits for a random time before forwarding the next packet. The protocols using RTS/CTS are not suitable for WSNs due to high overhead [29]. In order to show a true gain of exploiting the capture-effect, we purposely keep the MAC protocol simple. Figure 9 shows the numbers of links allocated to each channel by the Eavesdropping and utilizing the CTP protocol.

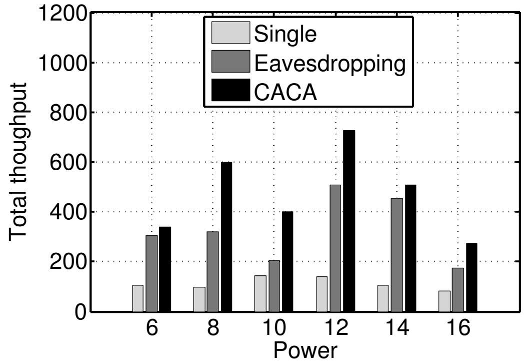


Fig. 10. Total number of packets received at base station when network experiences bursty traffic

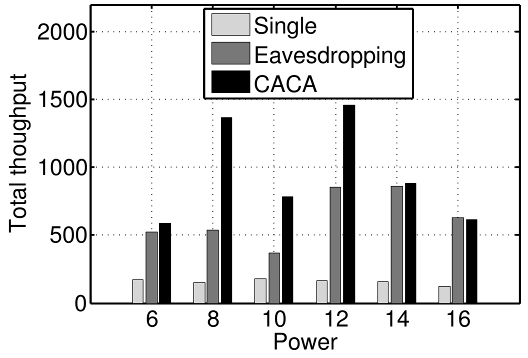
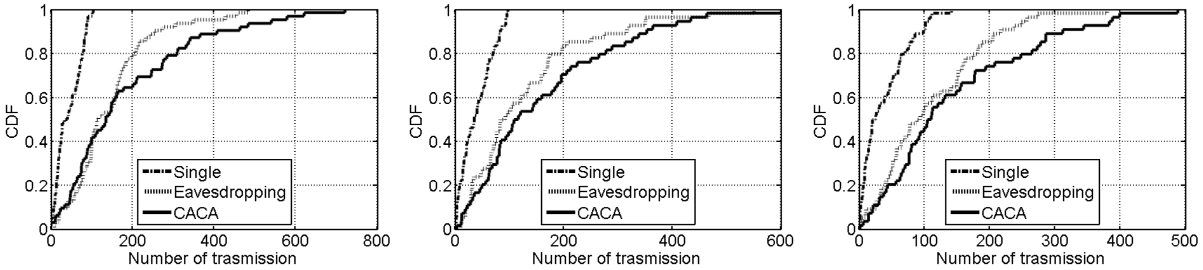


Fig. 11. Total number of packets received at base station when probability of concurrent transmission reduces by 50%

B. Experiment Results

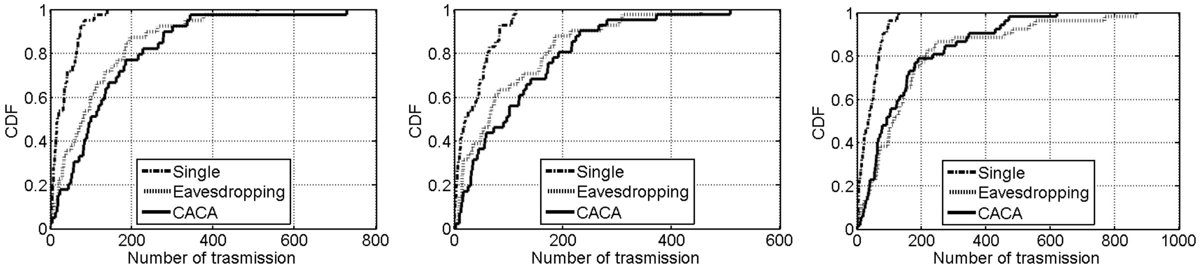
By comparing the results of a single channel network and Eavesdropping in Figure 10, we can see the performance gain of utilizing 3 channels without CACA. Our results show that utilizing 3 channels can provide performance gain up to nearly 4 times. The gains are generally higher when the



(a) TxPw = 6

(b) TxPw = 8

(c) TxPw = 10

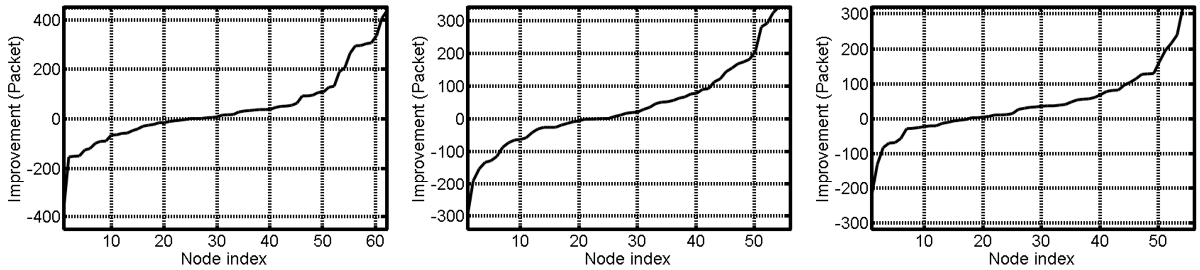


(d) TxPw = 12

(e) TxPw = 14

(f) TxPw = 16

Fig. 12. CDF showing the distributions of number of packet received by the parent nodes at different transmission powers (TxPw).



(a) TxPw = 6

(b) TxPw = 8

(c) TxPw = 10

Fig. 13. Impact of CACA at individual nodes

average number of neighbors increases (transmission power increases). This is expected since the network packet collision ratio is higher when the network density is high. Therefore, utilizing 3 channels reduces this collision and consequently it improves the network performance. Figure 10 shows that utilizing CACA always provides some additional performance gains on the top of Eavesdropping. Surprisingly, CACA can provide up to 100% additional improvement when overall link qualities and capture-effect probabilities in the network are high.

Although our previous results show 100% improvement, capture-effect is less effective in bursty traffic. In case of bursty traffic, many packets are lost due to many concurrent transmissions despite capture-effect. In order to simulate moderate traffic with less concurrent transmission, we reduce transmission probability. At transmission probability of 0.5, when a node has a packet to transmit it will delay its transmission about half of the time. In this case, we could observe up to 170% performance improvement by utilizing CACA compared to Eavesdropping in Figure 11.

Figure 12 shows the number of packets received by each parent node when the CTP tree is constructed using a single channel, Eavesdropping, or CACA. For example, Figure 12(a) shows that 90% of parent nodes received below 200 packets when the channels were assigned based on Eavesdropping. However, this reduces to 74% when the CACA is applied.

In the case of the transmission power to be higher than 12, the total throughput gains of CACA are below 50%. However, when we look at the performance gain from an individual node level, we can clearly see the benefit of CACA even in the case of high transmission powers. Figure 12(d) shows that the CDF of CACA always stays below Eavesdropping which indicates that almost all parent nodes have benefited from CACA. There is almost no gain from CACA when the transmission power is equal to 16 (neighbor density equal to 11). This is mainly due to packet loss from many concurrent transmissions despite the capture-effect. As shown in Figure 12(f), the CDFs of Eavesdropping and CACA are almost identical. In fact, the

Eavesdropping performed slightly better beyond 400 since the CACA could sometimes trade-off the link quality for better capture-effect probability. The effect of this poor trade off is shown in Figure 13. Figure 13 represents absolute gains or losses of CACA at individual nodes compared to Eavesdropping. Figure 13(a) shows that applying the CACA can help some nodes receive more than 400 additional packets while other nodes lose about 200 packets. This loss is due to the poor trade-off when the link quality and capture-effect qualities are generally low as shown in Figure 7(a) and Figure 7(e). But from Figure 13(b), we can clearly observe that higher proportion of nodes gained from CACA due to the improvement of link qualities and capture-effect qualities.

VI. CONCLUSION

In this paper, we design the channel assignment protocol for WSN which effectively exploits capture-effect. We first investigate the characteristics of capture-effect across different channels, times, and transmission powers. Then, we arrive to a conclusion that the capture-effect does not change as long as associated link quality level stayed steady. In order to realize this protocol, we develop a capture graph and a greedy graph coloring scheme for channel assignment. The capture graph directly represents the benefit of pair-wise capture-effect. The graph coloring scheme on the capture graph is a dual representation of channel assignment.

Extensive experiment is conducted on real a 140-node WSN testbed for performance evaluation. We compared our scheme to another similar channel assignment protocol designed for WSN. Performance evaluation shows our method outperforms this existing scheme by 100% in bursty WSN.

Our protocol is extremely useful in the case where the number of available channels is low. This really opens up a possibility of further exploiting the capture-effect for designing efficient multi-radio and multi-channel routing protocols

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2014006438), and also by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [10041244, Smart TV 2.0 Software Platform]. Note that Jaehoon (Paul) Jeong is the corresponding author.

REFERENCES

[1] "Wirelesshart specification," http://www.hartcomm2.org, accessed: 2015-12-14.

[2] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, "Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks," in *ICC'07*, 2007.

[3] Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *INFOCOM'08*, 2008.

[4] A. Bertossi, C. Pinotti, and R. Tan, "Channel assignment with separation for interference avoidance in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 222–235, March 2003.

[5] Y. Ding, K. Pongaliur, and L. Xiao, "Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 2, pp. 206–218, Feb 2013.

[6] A. Ghosh, O. Incel, V. Kumar, and B. Krishnamachari, "Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks," in *MASS'09*, 2009.

[7] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, "Flow-based real-time communication in multi-channel wireless sensor networks," in *EWSN'09*, 2009.

[8] G. Xing, M. Sha, J. Huang, G. Zhou, X. Wang, and S. Liu, "Multi-channel interference measurement and modeling in low-power wireless networks," in *RTSS'09*, 2009.

[9] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Improving link quality by exploiting channel diversity in wireless sensor networks," in *RTSS'11*, 2011.

[10] *Low-Cost Low-Power 2.4 GHz RF Transceiver*, Texas Instruments, 5 2009, rev. C.

[11] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Distributed channel allocation protocols for wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 9, pp. 2264–2274, Sept 2014.

[12] J. Lu and K. Whitehouse, "Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks," in *INFOCOM'09*, 2009.

[13] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *EmNetS-II'05*, 2005.

[14] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *SenSys'06*, 2006.

[15] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity," in *SIGCOMM'10*, 2010.

[16] S. Biswas and R. Morris, "Exor: Opportunistic multi-hop routing for wireless networks," in *SIGCOMM'05*, 2005.

[17] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *IPSN'11*, 2011.

[18] *2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver*, Texas Instruments, 3 2013, rev. C.

[19] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *NSDI'13*, 2013.

[20] O. Landsiedel, F. Ferrari, and M. Zimmerling, "Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale," in *SenSys'13*, 2011.

[21] Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *INFOCOM'08*, 2008.

[22] Y. Wu, M. Keally, G. Zhou, and W. Mao, "Traffic-aware channel assignment in wireless sensor networks," in *WASA '09*, 2009.

[23] X. Li, D. Wang, J. McNair, and J. Chen, "Residual energy aware channel assignment in cognitive radio sensor networks," in *WCNC'11*, 2011.

[24] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *SenSys '09*, 2009.

[25] J.-H. Yun and S.-W. Seo, "Novel collision detection scheme and its applications for {IEEE} 802.11 wireless {LANs}," *Computer Communications*, vol. 30, no. 6, pp. 1350 – 1366, 2007.

[26] S. Kosunalp, P. D. Mitchell, D. Grace, and T. Clarke, "Experimental study of capture effect for medium access control with aloha," *ETRI Journal*, vol. 37, no. 2, pp. 359 – 368, 2015.

[27] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *SECON '04*, 2004.

[28] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.

[29] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "Mmsn: Multi-frequency media access control for wireless sensor networks," in *INFOCOM '06*, 2006.

[30] J. Chen, Q. Yu, P. Cheng, Y. Sun, Y. Fan, and X. Shen, "Game Theoretical Approach for Channel Allocation in Wireless Sensor and Actuator Networks," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2332–2344, 2011.