PAPER Special Section on Fundamental Issues on Deployment of Ubiquitous Sensor Networks

Adaptive 3-dimensional Topology Control for Wireless Ad-hoc Sensor Networks

Junseok KIM[†], **Jongho SHIN**[†], *Nonmembers, and* **Younggoo KWON**[†], *Member*

SUMMARY Developing an adaptive 3-dimensional (3D) topology control algorithm is important because most wireless nodes are mobile and deployed in buildings. Moreover, in buildings, wireless link qualities and topologies change frequently due to various objects and the interference from other wireless devices. Previous topology control algorithms can suffer significant performance degradation because they only use the Euclidean distance for the topology construction. In this paper, we propose a novel adaptive 3D topology control algorithm for wireless ad-hoc sensor networks, especially in indoor environments. The proposed algorithm adjusts the minimum transmit power adaptively with considering the interference effect. To construct the local topology, each node divides the 3D space, a sphere centered at itself, into k equal cones by using Platonic solid (i.e., regular k-hedron) and selects the neighbor that requires the lowest transmit power in each cone. Since the minimum transmit power values depend on the effect of interferences, the proposed algorithm can adjust topology adaptively and preserve the network connectivity reliably. To evaluate the performance of algorithms, we conduct various experiments with simulator and real wireless platforms. The experimental results show that the proposed algorithm is superior to the previous algorithms in terms of the packet delivery ratio and the energy consumption with relatively low complexity.

key words: *3-dimension, Topology Control, Wireless Ad-hoc Sensor Network*

1. Introduction

Wireless ad-hoc sensor networking has emerged as one of the most researched areas because it has many special characteristics and unavoidable limitations compared with traditional wireless networks, such as cellular networks and wireless LANs. In wireless ad-hoc sensor networks, nodes communicate directly or along multi-hop paths to each other using wireless transceivers without the need for a fixed infrastructure such as base stations. Wireless sensor nodes need to establish ad-hoc networks in a self-organizing manner. The topology of wireless ad-hoc sensor networks is not static because wireless nodes move arbitrarily and the quality of wireless links varies frequently due to environmental changes. Wireless nodes, especially of wireless personal area networks or wireless sensor networks, are usually powered by small batteries, and they have limited memories and computing capacity. Network protocols, therefore, should be designed with consideration these unique characteristics and limitations.

The topology control is one of the energy-efficient wireless ad-hoc sensor networking techniques. Topology control protocols remove long and inefficient wireless links (i.e., select certain neighbors for communication), and adjust the transmit power, while maintaining the network connectivity to support energy-efficient routing, improve the channel capacity, and mitigate the medium access control (MAC)-level contention [1] [2]. The topology control has been heavily studied, but most of these studies assume that wireless nodes are distributed in a 2 dimensional (2D) plane [3]. In practice, however, wireless nodes are often deployed in a 3-dimensional (3D) space, such as multi-floor building, underwater, or underground. S. Poduri et al. showed that current 2D topology control algorithms are not able or require complex computations to apply in 3D networks [6]. Y. Wang et al. and S. Poduri et al. presented 3D topology control algorithms which are the 3D extension of the Yao graph (YG) and the cone based topology control (CBTC) respectively [4] [5].

These algorithms construct the topology locally and guarantee the network connectivity in a 3D space. In wireless ad-hoc sensor networks, where nodes are mobile, however, these algorithms incur large computation overhead because each node needs to repeat the topology construction procedure from beginning whenever a single neighbor moves or a single link quality changes. Moreover, these algorithms can degrade the network performance drastically due to the interferences, such as cement walls and/or signals of other wireless standard products, because they construct topologies by only using the Euclidean distance metrics. For example, wifi nodes (i.e., IEEE 802.11) can interfere with transmissions of IEEE 802.15.4-based sensor nodes. Although both 802.11 and 802.15.4 employ the carrier sense multiple access (CSMA) scheme to avoid the interference, 802.11 nodes often cannot detect signal of 802.15.4 nodes because the maximum transmission power (and default power for commercial usages) of 802.11 nodes is about 15dBm much higher than that of 802.15.4 nodes 0dBm. This problem gets worse when the topology control protocol of 802.15.4 nodes reduces the transmission power to save the energy [8].

To consider special characteristics and unavoidable limitations of wireless ad-hoc sensor network deployments, we translate them to a set of goals for topology control protocol:

- Interference robustness to minimize packet loss
- Low energy consumption to prolong network life

Manuscript received January 15, 2009.
Manuscript revised November 27, 2010.

[†]The authors are with Konkuk University, 1 Hwayang-dong, Gwangjin-gu, Seoul, 143-701, Korea. Email:(jskim1, jhshin, ygkwon@konkuk.ac.kr)

DOI: 10.1587/transcom.E93.B.1

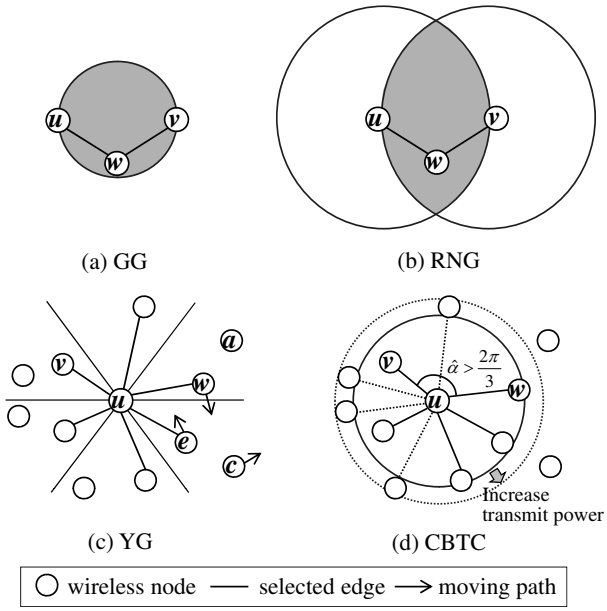


Fig. 1 Four most studied topology control algorithms.

- Low node degree to improve channel capacity
- Connectivity guarantee in 3D network
- Simple implementation for small sensor nodes

To meet these goals, we propose an adaptive 3D topology control algorithm for wireless ad-hoc sensor networks. The proposed algorithm determines the minimum transmit power by considering the interference effect. Each node u then divides the 3D space, a sphere centered at node u, into k equal cones by using a Platonic solid (i.e., a regular k-hedron) and selects the neighbor that requires the minimum transmit power in each cone. The proposed algorithm adjusts the minimum transmit power and the local topology adaptively in order to preserve the network connectivity and help routing protocols to forward messages reliably. To evaluate the performance of the proposed algorithm, we conduct various experiments with simulator and real wireless products. Simulation results show that the proposed algorithm can improve the channel capacity and requires fewer computations for adjusting the topology. Test-bed experiment results show that the proposed algorithm is superior to previous algorithms in terms of the delivery ratio and the energy consumption.

The rest of this paper is organized as follows. Section 2 describes related works on topology control algorithms. Then, we present the newly proposed 3D topology control algorithm and its protocol design in section 3 and 4 respectively. The experimental results and performance evaluations are given in section 5 and we conclude the paper in section 6.

2. Related Works

We consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a 3D space. Let $edge(u, v)$ be an one-hop wireless link between two nodes u and v. Let $G = (V, E)$ be a communication graph where E denotes the set of edges. Topology control algorithms remove redundant edges and keep energy efficient edges to prolong the network lifetime. Let $G_{TC} = (V, E_{TC})$ be the constructed communication graph with a topology control (TC) algorithm. Here, we review four most studied topology control algorithms: the Gabriel graph (GG), the relative neighbor graph (RNG), the Yao graph (YG), and the cone-based topology control (CBTC) [1]. Let $\|u, v\|$ denote the Euclidean distance between two nodes u and v. GG includes $edge(u, v)$ if and only if (iff) the circle with diameter $\|u, v\|$ contains no other node. RNG includes $edge(u, v)$ iff the intersection of two circles centered at two nodes u and v and with radius $\|u, v\|$ contains no other node. For example, shown in Fig. 1 (a) and (b), $edge(u, v)$ is not included in both E_{GG} and E_{RNG} and instead $edge(u, w)$ and $edge(w, v)$ are included in.

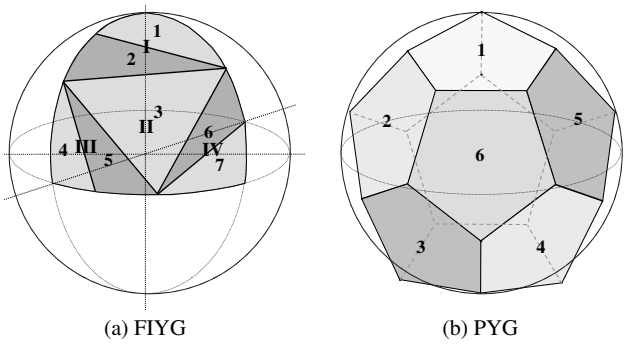


Fig. 2 Comparison of two cone division methods.

In YG, each node u divides the region, a circle centered at node u, into equal $k \geq 6$ cones and chooses the shortest edge in each cone. Let $C(u, v)$ denote the cone of node u containing node v in YG. In CBTC, each node u broadcasts a HELLO message, including its location information, with the minimum transmit power and collects replies. If there is an empty cone with apex angle $\alpha > 2\pi/3$, the node u increases the transmit power and broadcasts the HELLO message again. For example, shown in Fig. 1-(d), node u increases the transmit power since the angle $\hat{\alpha} = \angle vuw$ (i.e., the acute angle between two lines $\|v, u\|$ and $\|u, w\|$) is larger than $2\pi/3$. If there is no empty cone, node u selects $edges(u, v)$ for all neighbors v within the transmit range. CBTC looks similar to YG because both use the concept of cone for the topology construction. However, CBTC requires much more computations for adjusting the topology than YG when nodes are mobile because the position of cones is fixed in YG but it is not in CBTC. Consider that a node v moves. In CBTC, neighbors of the node v need to reset the edge set and restart the topology construction, but in YG, they simply reselect another neighbor in one or two cones only if necessary. For example, as shown in Fig. 1-(c), in YG, node u does not change the topology when node e and c have moved because node e is still the closest neighbor of node u in $C(u, e)$. Node u, whereas, has to reselects an edge in $C(u, a)$ when node w has moved. In CBTC, when

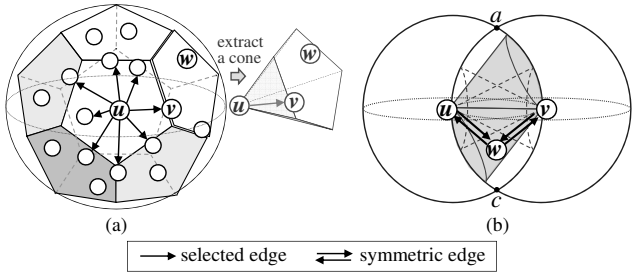


Fig. 3 Neighbor selection with dodecahedron (a) and proof of RNG ⊆ PYG for $k \geq 8$ (b).

node w moves, node u has to check whether $\angle wux$ is larger than $2\pi/3$ for all neighbors x within the transmit range

The topology control has been heavily studied in recent 10 years, but there are not enough studies in 3D topology control. S. Poduri et al. proposed the neighbor-every-theta (NET), a 3D extension of CBTC [5]. In NET, each node computes the Delaunay Triangulation (DT) on a sphere (centered at itself) to determine the existence of empty 3D cones with apex angle $\pi > \alpha$. NET is not efficient in practice since the expense of DT construction is very high and each node has to re-compute DT whenever a neighbor moves [3]. Y. Wang et al. [4] presented two 3D topology control algorithms. The first algorithms, named the fixed YG (FIYG), divides a sphere into 32 or 56 cones as shown in Fig. 2-(a) (i.e., 1/8 sphere is divided into 4 or 7 pieces). FIYG selects too many edges and hence the MAC-level contention does not reduce much. Since the shape of cones is not equal, FIYG selects more neighbors in a specific region, and thus network traffic can be congested in this region. In the second algorithm, named the flexible YG (FLYG), each node u selects an $edge(u, v)$ iff there is no other node w closer than node v in a 3D circular cone with apex angle $\alpha = \pi/2$ (steradian) and axis line $u \rightarrow v$. FLYG bounds the number of selected edges by 26, much less than FIYG, but the position of 3D cones is unfixed. When a neighbor moves, therefore, FLYG needs to reset the edge set and restart the topology construction from the beginning. That is, FLYG requires a lot of computations for adjusting the topology as CBTC does.

3. 3D Topology Control With Platonic Solid

In this section, we propose an YG-based topology control with Platonic solid (PYG) for 3D wireless ad-hoc sensor networks. In PYG, each node broadcasts a HELLO message including its location information with the maximum transmit power. After collecting HELLO messages, each node u divides the 3D space, a sphere centered at node u, into k equal cones by using a Platonic solid (i.e., a regular k-hedron) and selects the neighbor with the lowest link cost (e.g., the shortest distance or the lowest minimum transmit power) in each cone. For example, shown in Fig. 3-(a), node u divides its 3D space into 12 equal cones by using a dodecahedron and selects the closest neighbor in each cone. (It seems that node u selects more than one neighbor in some cones because 3D cones overlap on figure.) After the neighbor selection, each node u broadcasts its selected neighbor set, ${}^{u}N$. When a node u is received ${}^{v}N$ from node v, it compares two selected neighbor sets (i.e., ${}^{u}N$ and ${}^{v}N$) and inserts certain edges into two edge sets: default edge set, E_{PYG}, and symmetric edge set, E_{SPYG}. Both sets are defined as follows:

$$\begin{aligned} edge(u,v) \in E_{PYG} \text{ iff. } v \in {}^{u}N \text{ or } u \in {}^{v}N \\ edge(u,v) \in E_{SPYG} \text{ iff. } v \in {}^{u}N \text{ and } u \in {}^{v}N \end{aligned} \quad (1)$$

That is, an $edge(u, v)$ is remained in $G_{PYG} = (V, E_{PYG})$ if more than one of two nodes u and v select the other node, whereas it is remained in G_{SPYG} iff both node u and v select each other. Here, SPYG denotes the symmetric YG with Platonic solid and is similar to the symmetric Yao graph in 2D [2].

G_{PYG} is connected for all Plastic solids (i.e. $k \geq 4$). For example, shown in Fig. 3-(a), although node w is not selected by node u (and by other nodes), it can select node u (and/or other nodes) and connect itself to the network. However, G_{SPYG} is connected when using an octahedron, dodecahedron, and icosahedrons (i.e. for $k \geq 8$). This result can be proved by comparison with RNG as shown in Fig. 3-(b). It is well known that RNG guarantees the connectivity of 3D wireless ad-hoc networks [5]. In RNG, an $edge(u, v)$ is remained if a region, looks like a Rugby ball, does not contain any other node, where the region is formed as the intersection of two spheres centered at two nodes u and v, and with radius $\|u, v\|$ (Refer Fig. 1-(b)). SPYG includes an $edge(u, v)$ if nodes u and v have the lowest link cost in $C(v, u)$ and $C(u, v)$ respectively (e.g., two nodes u and v are closest to each other in each 3D cones). That is, in SPYG, an edge(u, v) is remained if a region, formed as the union of two 3D cones with slant height $\|u, v\|$, does not contain any other nodes. Note that the angle $\angle avc = \pi$ steradian in Fig. 3-(b). The Rugby-ball-like region of RNG encloses two 3D cones of SPYG if the apex angle of the cones is equal or smaller than $\pi/2$. The apex angle of the 3D cone made by a regular k-hedron is $4\pi/k$ steradian. Therefore, the region (where $edge(u, v)$ is removed if other node w is located in) of RNG can enclose that of SPYG if $k \geq 8$. This means that the edge set of RNG includes that of SPYG for $k \geq 8$. Since RNG guarantees the network connectivity, it is proven that G_{SPYG} is connected for $k \geq 8$

4. Adaptive 3D Topology Control Protocol

Previous topology control algorithms use Euclidean distance as the link cost for the topology construction. These algorithms can suffer significant packet losses in indoor environments since link qualities are affected by several objects such as cement wall or iron door. In addition, 2.4GHz ISM frequency band is heavily crowded by several wireless standard devices such as IEEE 802.11 WLAN, IEEE 802.15.1 Bluetooth, and IEEE 802.15.4 ZigBee; therefore

link qualities fluctuate heavily and the network performance is degraded drastically due to the interference of other wirelessstandard devices. In this section, we propose an adaptive 3D topology control protocol, namely the adaptive Yao graph with Platonic solid (A-PYG). A-PYG determines the minimum transmit power and adjusts the local topology adaptively with considering the interference effect to preserve the network connectivity and help a routing protocol forward messages reliably in indoor environments.

Recent experimental study shows that the signal to interference plus noise ratio (SINR) is a good link quality indicator for the transmit power control and the topology control [8]. Wireless transceivers are able to receive messages reliably when SINR value is over a certain threshold regardless of the interference effect (See Appendix). SINR is expressed in milliwatt as follows:

$$SINR = P_S / (P_I + P_N) \tag{2}$$

where P_S, P_N and P_I are the desired signal power, the noise power and the interference power. We can derive the signal power threshold (in decibel), P_{S-TH}, for a successful reception as follows:

$$P_{S-TH} = 10log\,(S_{TH}\,(P_I + P_N)) \tag{3}$$

where S_{TH} is the SINR threshold (in milliwatt) for a successful reception. The minimum transmit power from node u to node v, ${}^{min}P_T(u,v)$, is determined (in decibel) as follows:

$${}^{min}P_T(u,v) = PL(u,v) +^{v} P_{S-TH} \tag{4}$$

where $PL(u,v)$ is the path loss between two nodes u and v (in decibel), and ${}^{v}P_{S-TH}$ is the signal power threshold value of node v. The noise power depends on the hardware and can be set manually [8] [10]. The interference power can be measured by a transceiver. For example, with CC2420 IEEE 802.15.4-compatible transceiver, we can measure the interference power by reading the received signal strength indicator (RSSI) register and distinguish the interference signal from the desired signal by checking the start of the frame delimiter (SFD) pin [10]. (The SFD pin goes high when the transceiver receives the IEEE 802.15.4 signal.) SINR threshold can be derived from the bit error ratio (BER) model. For example, the BER model of IEEE 802.15.4 as follows [7]:

$$P_B = \frac{8}{15} \cdot \frac{1}{16} \cdot \sum_{k=2}^{16} -1^k \binom{16}{k} e^{20 \cdot SINR \cdot \left(\frac{1}{k}-1\right)} \tag{5}$$

In experiments, we set the SINR threshold value as 0.4dBm to satisfy the IEEE 802.15.4 receive sensitivity requirement (99% packet reception ratio (PRR) with 100bytes packet).

The minimum transmit power, determined by (4), depends on distance, objects between nodes, and the interference power at the receiver. A-PYG uses the minimum transmit power value as the link cost for the topology construction and therefore it can remove wireless links corrupted by objects and/or interferences. In the initialization phase of A-PYG, each node u measures the interference power and determines the signal power threshold value, ${}^{u}P_{S-TH}$, by using (2). Each node then broadcasts a HELLO message including its location and the signal power threshold value with the maximum transmit power ${}^{max}P_T$. When a node receives a HELLO message, it measures the signal power, P_S, and estimates the path loss, $PL =^{max} P_T - P_S$. Then, the minimum transmit power value, ${}^{min}P_T$, is determined by using (4). After determining ${}^{min}P_T$ values for neighbors, each node u selects the neighbor that requires the lowest minimum transmit power in each cone and then broadcasts the selected neighbor set, ${}^{u}N$. With the selected neighbor sets of neighbors and itself, each node constructs its local topology by using (1).

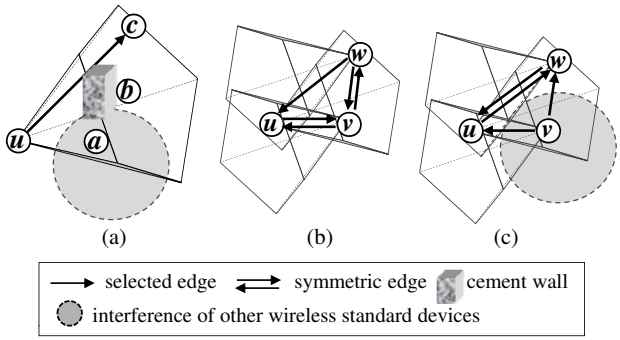


Fig. 4 Example of topology adjustment (a), and topology of A-PYG under no interference effect (b) and under interference effect (c)

In the operation, each node measures the interference power periodically. When the signal power threshold value of a node u is changed much because of the interference effect, the node immediately broadcasts a HELLO message including the updated threshold value. Then, neighbors of the node u adjust the minimum transmit power (for the node u) and the selected neighbor set if necessary. If the selected neighbor set is changed, the node broadcasts it to let neighbors adjust the topology. The path loss and the minimum transmit power values are also updated by receiving periodic HELLO messages and/or overhearing other messages. Figure 4-(a) illustrates how A-PYG adjusts the topology adaptively under the effect of interferences. When there is no interference, node u selects the closest neighbor a since ${}^{min}P_T(u,a)$ value is lowest in the cone, $C(u,a)$. When node a is affected by strong interference of other wireless standard devices, it increases the signal power threshold value and broadcasts the updated threshold value. Node b is closer to node u than node c, but the path loss of $edge(u,b)$ is much larger than that of $edge(u,c)$ because of a thick cement wall. As a consequence, node u increases ${}^{min}P_T(u,a)$ value and changes its neighbor selection from node a to node c. A-PYG can remove unstable wireless links affected by interferences and therefore help a routing protocol to forward messages reliably. When only bidirectional links are used (i.e., in A-SPYG), however, nodes can be disconnected from wireless ad-hoc network in the worst case as shown in Fig.

Algorithm 1 Find a cone that contains a neighbor v

```
# In a view of node u
# χ_i is relative coordinates of base center of i-th cone
dist = ∞
for i = 1 to k do
    if dist > ||v, u + χ_i|| then
        dist = ||v, u + χ_i||
        cone = i
    end if
end for
return cone
```

Algorithm 2 Adjust local topology when neighbor v's location or ${}^{min}P_T(u,v)$ changes

```
# In a view of node u
# w_{C_i} is neighbor with lowest min P_T in i-th cone
if node v moves in same cone C_a then
    if v ≠ w_{C_a} & min P_T(u,v) < min P_T(u, w_{C_a}) then
        remove edge(u, w_{C_a})
        select edge(u, v)
        w_{C_a} = v
    end if
else
    if v = w_{C_a} then
        remove edge(u, v)
        find min P_T in C_a
    end if
    if min P_T(u,v) < min P_T(u, w_{C_b}) then
        remove edge(u, w_{C_b})
        select edge(u, v)
        w_{C_b} = v
    end if
end if
```

4-(b) and (c). To prevent this problem, when a node uses A-SPYG and fails to establish the route to the destination node, it uses A-PYG temporarily.

In YG, each node determines that which cone a neighbor is located in by using angle information in a 2D plane. In PYG, whereas, it is not easy to find a cone that contains a neighbor by using angle information. Here, we present a simple method to find - which cone a neighbor is located in - with the distance information. Let χ_i be the location of the base center of the i-th 3D cone (i.e., the center of the i-th face of a Platonic Solid). Each node can determine which cone contains neighbor v by comparing $\|v,\chi_i\|$ values. For example, if a neighbor v is located in the second 3D cone, $\|v,\chi_2\|$ will be the smallest value (i.e., $\|v,\chi_2\| < \|v,\chi_i\|$ for i =1, 3, ..., and k). Algorithm 1 shows this cone determination procedure of A-PYG. Center coordinates of each face of a Platonic solid can be found in Appendix.

In A-PYG, position of cones of a node is fixed unless the node moves. This means that A-PYG can adjust the local topology with a few computations by re-selecting a neighbor in one or two cones when a neighbor moves and/or a link quality changes. As we explained before, topology control algorithms – of which the position of cones is unfixed (such as FLYG and NET) – require a lot of computations for adjusting the local topology. Algorithm 2 shows the pseudocode of the topology adjustment of A-PYG.

Table 1 Parameter Values in Experiments

Parameter	Value	Parameter	Value
d_0	1m	T_{LIFS}	0.64ms
PL_0	40dB	T_{SIFS}	0.192ms
η	2.7	T_{ACK}	0.192ms
σ	2dB	T_{BO}, P_N	variable
M	3dB	E_T	25 52mW
${}^{max}P_T$	0dBm	E_R	59.1mW

5. Experimental Results

Firstly, we use a simulator to evaluate the performance of algorithms in ideal free space. Then, we conduct field test to understand the interference impact on the performance.

5.1 Simulation Results

We compare the performance of the proposed algorithm (PYG) with that of the FIYG and FLYG algorithms in large scale wireless ad-hoc sensor networks by using the MATLAB-based simulator. In simulations, we use the log-distance path loss model. The path loss at a distance d is defined in dB scale as follows:

$$PL(d) = PL_0 + 10\eta log10(d/d_0) + X_\sigma \quad (6)$$

where PL_0 is the path loss at the close-in reference distance d_0, η is the path loss exponent, and X_σ is a zero mean Gaussian distributed random variable with standard deviation σ. We measure above values by using TELOSB motes which have CC2420 IEEE 802.15.4 compatible transceiver [10] [12]. Table 1 shows parameter values used in simulations. In simulations, we place from 25 to 150 wireless nodes randomly in 500x500x500 space where there is no interference or obstacle. Every node collects neighbor information and then establishes the local topologies by using six different topology control algorithms: $FIYG_{k=32}$, $FIYG_{k=56}$, FLYG, $PYG_{k=8}$, $PYG_{k=12}$, and $PYG_{k=20}$.

Figure 5 shows the average node degree of the algorithms as a function of the number of nodes. The node degree means the sum of in- and out-degree of node u; and the in-degree and the out-degree are the number of neighbors that select node u and the number of neighbors that is selected by node u respectively. YG-based algorithms, which use position-fixed cones, (such as PYG and FIYG) can bind the node out-degree by k because they divide the 3D space, a sphere centered at itself, into k cones and select a neighbor in each cone. PYG has much lower node degree than FIYG because k values of PYG (4, 6, 8, 12, and 20) are much smaller than that of FIYG (32 and 56). From Fig. 5, we can see that the node degree of $PYG_{k=8}$ and $FIYG_{k=56}$ are lowest and highest respectively. The node degree of FLYG is slightly larger than that of $PYG_{k=8}$. The reason is that both $PYG_{k=8}$ and FLYG use 3D cones with apex angle $\pi/2$; however 3D cones of FLYG do not have the fixed position and they intersect with each other, and hence FLYG can have more edges than $PYG_{k=8}$. The node degree values of PYG

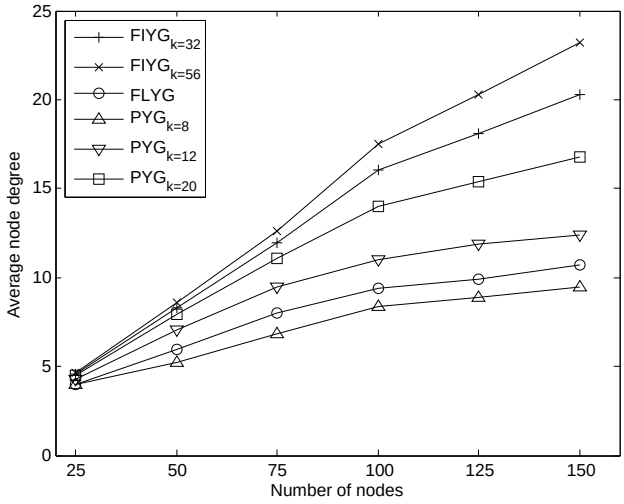


Fig. 5 Average node degree of algorithms

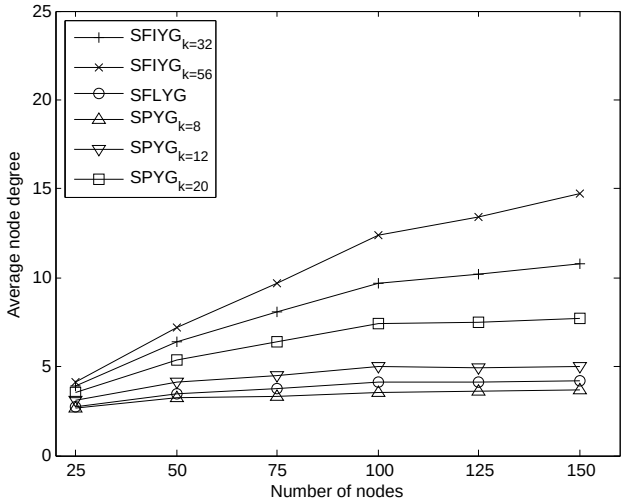


Fig. 6 Average node degree of algorithms with bidirectional edges only

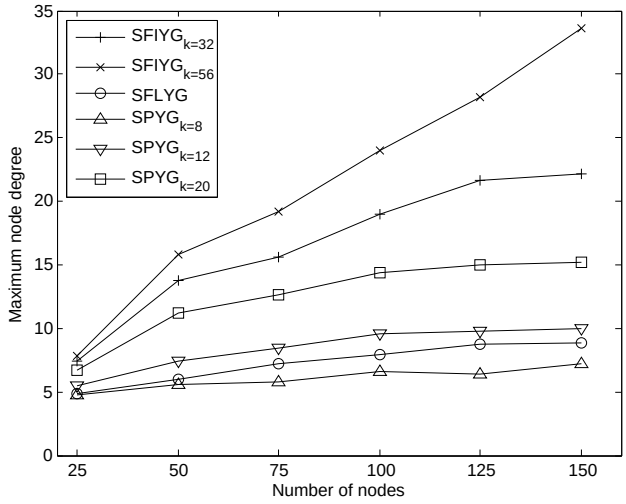


Fig. 7 Maximum node degree of algorithms with bidirectional edges only

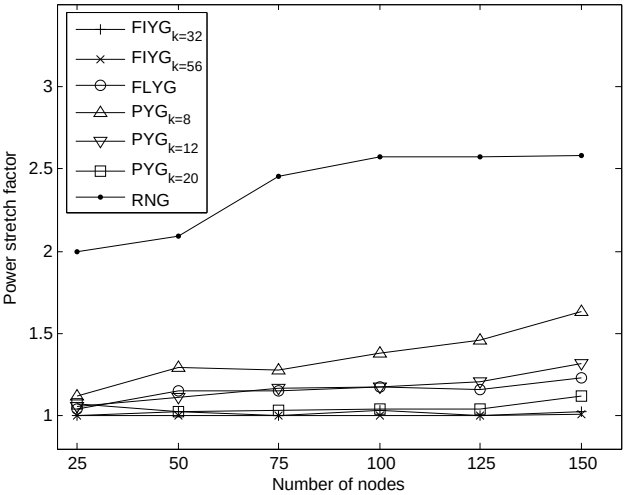


Fig. 8 Power stretch factor of algorithms

and FIYG increases gradually as the number of nodes increases and are not bounded by k. The reason is that the node out-degree is bounded by k but the node in-degree is not bounded by k.

The node in-degree can be reduced much by removing uni-directional edges (i.e. by constructing the symmetric graph). Figure 6 and 7 show the average and maximum node degree of the algorithms when only bidirectional edges are used. SFIYG and SFLYG denote the symmetric FIYG and the symmetric FLYG respectively (i.e., both consists of bidirectional edges). We can see that constructing a symmetric topology graph can bound the maximum node degree by k and reduce the average node degree by about 50 percent.

In the most common propagation model, the transmission power needed to support a link (u, v) is $\|u, v\|^\beta$ where β is a real constant between 2 and 5 dependent on the environment [2]. Let $p_{TC}(u, v) = \sum_{i=1}^{h} \|w_{i-1}, w_i\|^\beta$ be the total transmission power by the shortest path from node u $(= w_0)$ to node v $(= w_h)$ in a communication graph constructed by the topology control (TC) algorithm. The power stretch factor ρ is then defined as follow:

$$\rho = \max(p_{TC}(u, v)/p(u, v)) \text{ for nodes } u, v \in V \qquad (7)$$

where $p(u, v)$ denotes the total transmission power by the shortest path between node u and node v in networks without topology control algorithm. Most topology control studies use this power stretch factor to evaluate the energy performance of topology control algorithms [1] [2].

Figure 8 and 9 show the average power stretch factor of algorithms, when unidirectional edges are used and removed respectively, as a function of the number of nodes. We can see that the power stretch factor values of PYG and FIYG decrease as the number of cones increases. The power stretch factor values of PYG therefore are higher than that of FIYG; but it is fairly low compared to RNG which is

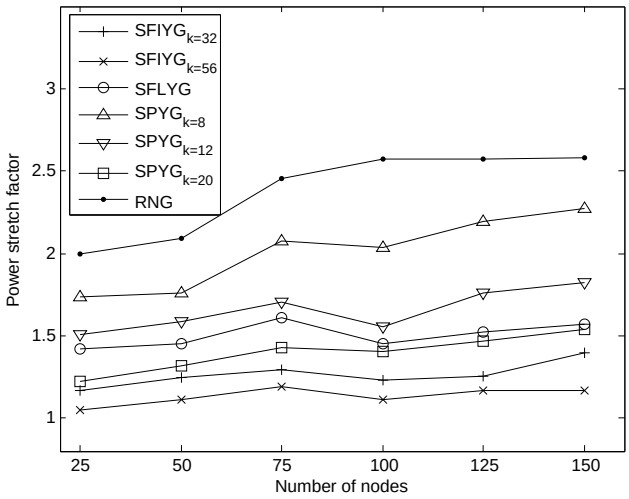


Fig. 9 Power stretch factor of algorithms with bidirectional edges only

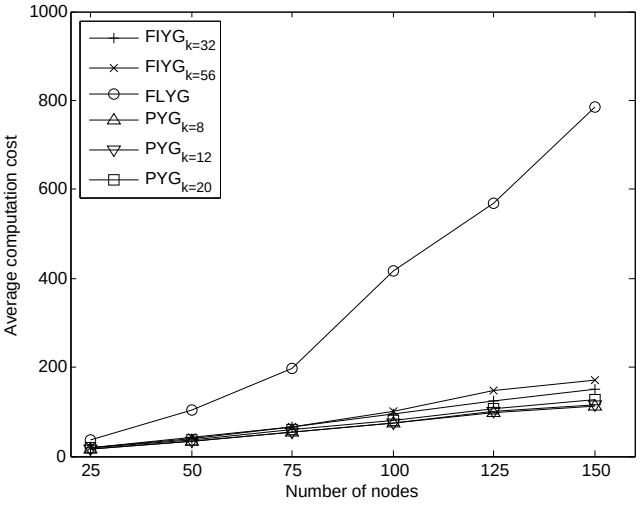
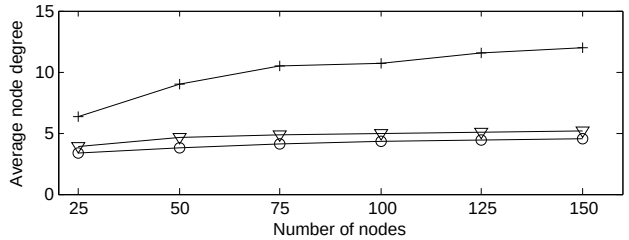


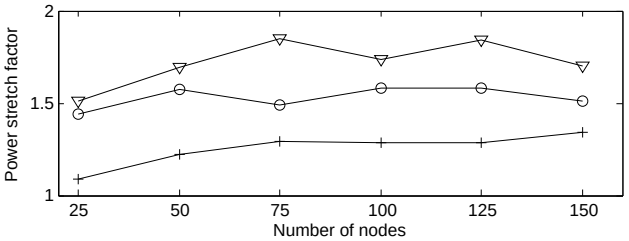
Fig. 10 Average computation cost of algorithms

used for the efficient broadcasting [2]. As shown in Fig. 9, the power stretch factor values are increased by about 35 percent when algorithms remove the unidirectional edges. From Fig. 8 and 9, we can say that there is a trade-off between the node degree and the power stretch factor. The power stretch factor values of PYG are larger than that of FIYG because PYG selects fewer edges than FIYG. That is, FIYG can consume less energy than PYG in the free space. However, in the later section, test-bed experiment results show that PYG is more energy efficient than other algorithms (including FIYG) because the other algorithms suffer high packet losses and waste a lot of energy for retransmissions due to the interference. The power stretch factor of FLYG is similar to that of PYG$_{k=12}$. FLYG shows relatively good performance in terms of the node degree and the power stretch factor; however it requires a lot of computations for adjusting the topology.

As we mentioned before, when a neighbor moves



(a) Average node degree of algorithms



(b) Average power stretch factor of algorithms

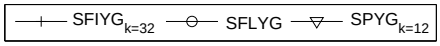


Fig. 11 Performance evaluation in non-uniformly distributed scenario

or a link quality changes, YG-based algorithms that use position-fixed cones (such as PYG and FIYG) can adjust the local topology with a few computations by reselecting a neighbor in one or two cones if necessary. However, topology control algorithms of which the position of cones is not fixed (such as FLYG and NET) need to reset the edge set and reconstruct the topology from the beginning when a neighbor moves or a link quality changes. Figure 10 shows the average computation cost for the topology adjustment as a function of the number of nodes. In simulations, each node moves within 15m in an arbitrary direction with a probability 30%. That is, about 30% of wireless nodes are moving randomly within 15m. As shown in Fig. 10, the computation cost of FLYG is much more than that of PYG and FIYG and increases exponentially as the number of nodes increases. The first reason is that, the complexity of the initial neighbor selection of FLYG is $O(m(\log m + 1))$ where m denotes the number of neighbors, whereas that of PYG and FIYG is $O(m)$. The second reason is that FLYG needs to reset the edge set and restart the neighbor selection procedure from beginning when a single neighbor moves, whereas PYG and FIYG reselect a neighbor in one or two cones if necessary. In practice, wireless nodes of wireless ad-hoc sensor networks usually have very limited computation capacity and hence high complexity of the topology construction and adjustment is a critical disadvantage in terms of the implementation.

Lastly, we placed wireless nodes randomly with Rayleigh distribution to evaluate the performance of algorithms in non-uniformly distributed scenario. That is, a specific area is crowded by wireless nodes but there are a few wireless nodes in other area. Figure 11(a) and (b) show that the node degree and the power stretch factor of algorithms

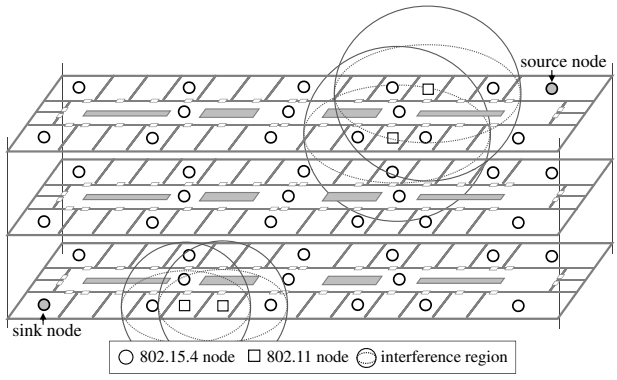


Fig. 12 Testbed experiment scenario

do not change much compared to when wireless nodes are deployed randomly with uniform distribution.

5.2 Testbed Experimental Results

In section 4, we present a novel adaptive 3D topology control protocol, named as A-PYG (or A-SPYG), that adjusts the local topology adaptively to environmental changes. To evaluate the performance of A-PYG, we conduct several experiments with forty-five 802.15.4 nodes and four 802.11 nodes in the three stories building as shown in Fig. 12. The 802.15.4 nodes have CC2420 IEEE 802.15.4 transceiver which support data rate of 250kbps [10]. The 802.11 nodes have RT2870 IEEE 802.11a/b/g/n transceiver which supports data rate up to 300Mbps [11]. Both transceivers operate in 2.4GHz ISM frequency band. We set the frequency channel of 802.15.4 and 802.11 nodes as 22 and 11 respectively to make two frequency bands be overlapped. We set the interference occupy ratio (IOR) as 0.2 (see Appendix). In experiments, each 802.15.4 node constructs the local topology by using the A-SPYG$_{k=12}$. After constructing the topology, the 802.15.4 source node tries to build a route, over the constructed topology, to the 802.15.4 destination node with the ad-hoc on-demand distance vector (AODV) routing agent [13]. The 802.15.4 source node then transmits 50-byte messages to the 802.15.4 sink node through the route at the rate of 1 packet per second. In the meanwhile, one of two 802.11 nodes downloads a big file from another by using the file transfer protocol (FTP) program. We set the transmit power of 802.11 nodes manually from -15 to 10dBm for each experiment. We repeat the same experiments by using three other topology control algorithms: SFIYG$_{k=32}$, SFLYG, and NET. As we mentioned before, previous topology control studies assume that the transmission power (i.e., $\|u,v\|^{\beta}$) is a dominant factor in the energy consumption; however, in reality, wireless nodes consume much energy in other procedures such as receiving messages or exchanging acknowledge packets. To better evaluate the energy performance of algorithms, we use the following energy mode:

$$E = E_T \times T_{DATA} + E_R \times (T_{LIFS} + T_{BO} + T_{SIFS} + T_{ACK}) \quad (8)$$

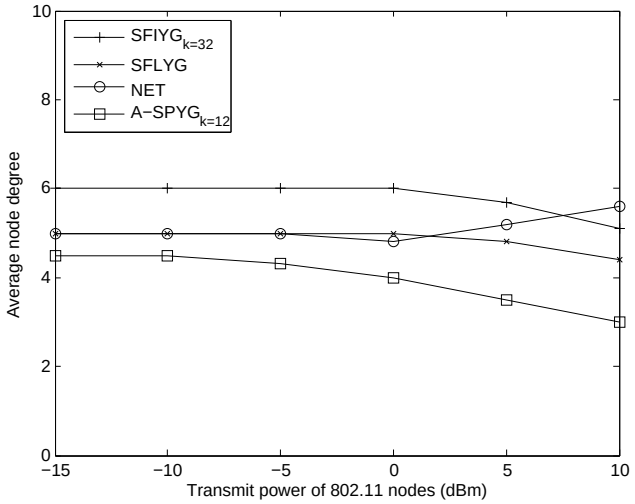


Fig. 13 Average node degree of algorithms

where E_T and E_R denote the energy consumption in milliwatt in the transmit mode and the receive mode respectively. T_{DATA}, T_{ACK}, T_{LIFS}, T_{SIFS}, and T_{BO} denote the duration of the data packet, the acknowledge packet, the long inter-frame space, the short inter-frame space, and the backoff respectively [7]. Table 1 shows the above parameter values used in the energy consumption calculations.

Figure 13 shows the average node degree of algorithms as a function of the transmit power of 802.11 nodes. Note that signal from 802.11 nodes is the interference to 802.15.4 nodes. The node degree of A-SPYG decreases when the transmit power of 802.11 nodes increases over -10dBm. The reason is that A-SPYG does not select any neighbor in a cone when all neighbors in the cone are strongly affected by the interference and hence the lowest minimum transmit power in the cone is increased over the maximum transmit power. The node degree values of SFIYG, SFLYG, and NET are relatively unchanged compared to A-SPYG because they only use Euclidean distance for the topology construction and do not consider the interference effects. The node degree of SFIYG and SFLYG is decreased when the transmit power of the 802.11 nodes increases over -0dBm. The reason is that 802.15.4 nodes near 802.11 nodes cannot receive HELLO messages due to the heavy interference, and hence some 802.15.4 nodes have fewer neighbors that can communicate than when there is no interference. The node degree of NET is increased when the transmit power of 802.11 nodes increases over 0dBm because the heavy interference yields a big area where 802.15.4 signal cannot be received. In NET, therefore, some 802.15.4 nodes have to increase the transmit power and select more edges to reduce the apex angle of big empty 3D cones yielded by the heavy interference of 802.11 nodes.

Figure 14 and 15 show the delivery and energy performance of algorithms when using AODV as a routing agent. NET shows the worst delivery and energy performance. The first reason is that NET constructs the topology based on the

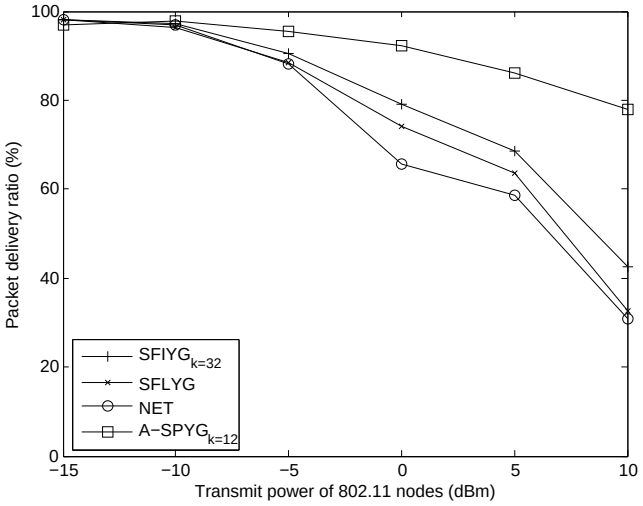


Fig. 14 Delivery performance of algorithms

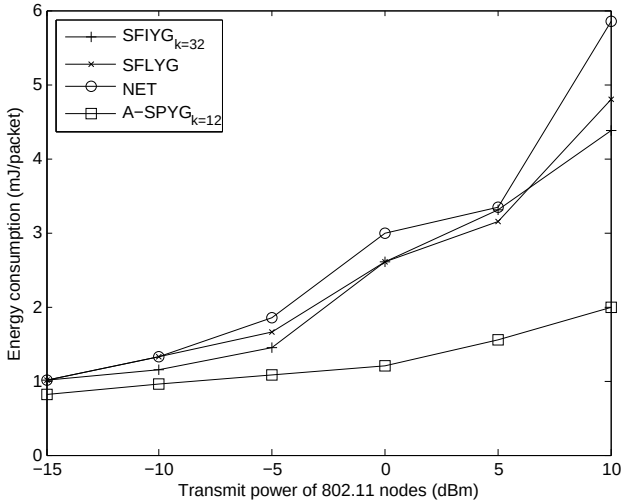


Fig. 15 Energy performance of algorithms

location information and hence the route on the constructed topology lies across the interference region. The second reason is that, in NET, 802.15.4 nodes near 802.11 nodes (i.e., near and within the interference region) can select long and low-quality wireless links to reduce the apex angle of big empty 3D cones made by the interference of 802.11 nodes. SFIYG and SFLYG also suffer high packet losses because they build the topology based on the location information and do not take into account the effect of 802.11 interference and obstacles. A-SPYG shows the highest delivery and energy performance. The reason is that A-SPYG considers the interference effects of 802.11 and obstacles in the neighbor selection and it does not selects a neighbor in a cone when all neighbors in the cone suffer heavy interference effects. The routing agent, therefore, can forward messages through other paths which are not affected by the interference. Since A-SPYG does not waste the energy for retransmissions, it outperforms other algorithms in terms of the energy consumption. From various experimental results, we can say that the proposed 3D topology control algorithm is able to improve the network performance and help wireless applications work properly in real deployments.

6. Conclusions

This paper presented the adaptive 3D topology control algorithm for wireless ad-hoc sensor networks. Our main goal is to develop an YG-based 3D topology control algorithm that is able to build energy-efficient topology and adjust the topology adaptively to environmental changes, especially in buildings where various communication obstacles, such as cement wall or interference of other standard devices, exist. To evaluate the performance, we implement algorithms in MATLAB simulator and real wireless products, and conduct various experiments. From simulation results, the proposed algorithm shows that it has low node degree (which means low MAC-level contention and high channel capacity) and requires fairly low computation cost for the topology adjustment. Testbed experiment results confirm that the proposed algorithm is superior to previous algorithms (FIYG, FLYG, and NET) in terms of the delivery and the energy performance when there are strong interference effects.

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. R01-2008-000-20109-0 and No. 2009-0089304).

References

[1] P. Santi, “Topology Control in Wireless Ad Hoc and Sensor Networks”, John Wiley & Sons, Ltd., 2005.
[2] X. Li, “Algorithmic, geometric and graphs issues in wireless networks”, Wirel. Commun. Mob. Comput. Vol. 3, pp. 119-140, Mar., 2003.
[3] Y. Wang, L. Cao, T. Dahlberg, F. Li, and X. Shi, “Self-organizing fault-tolerant topology control in large-scale three-dimensional wireless networks”, ACM Trans. on Autonomous and Adaptive Systems, vol. 4, no. 3, Jul., 2009.
[4] Y. Wang, F. Li, T. Dahlberg, “Power efficient 3-dimensional topology control for ad hoc and sensor networks”, Proc. GLOBECOM, San Francisco, CA, USA, Nov. 27-Dec. 1, 2006.
[5] S. Poduri, S. Pattem, B. Krishnamachari, and G. Sukhatme, “Using local geometry for tunable topology control in sensor networks”, IEEE Trans. on Mob. Comput., vol. 8, no. 2, Feb. 2009.
[6] S. Poduri, S. Pattem, B. Krishnamachari, and G. Sukhatme, “Sensor Network Configuration and the Curse of Dimensionality”, Proc. EmNets, Harvard University, Cambridge, MA, USA, May 30-31, 2006.
[7] IEEE Std 802.15.4, “Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)”, Oct., 2003.
[8] J. Kim and Y. Kwon, “Interference-aware topology control for low rate wireless personal area networks,” IEEE Trans. on Consumer Electronics, vol. 55, no. 1, pp. 97-104, Feb., 2009.
[9] K. MacLean, “A geometric analysis of the platonic solids and other semi-regular polyhedral”, Loving Healing Press, 2007.

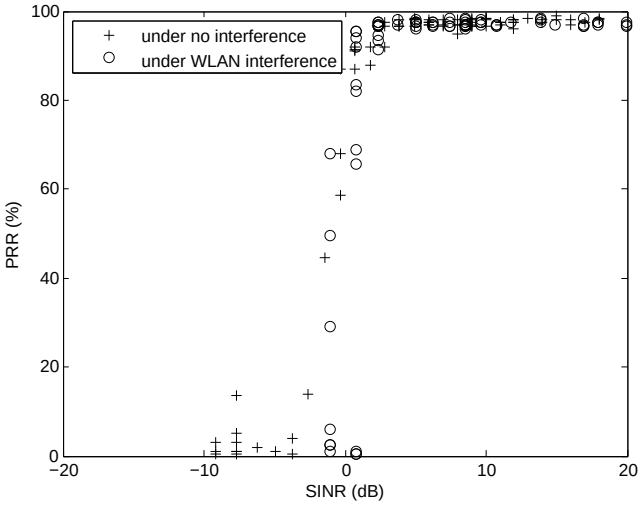
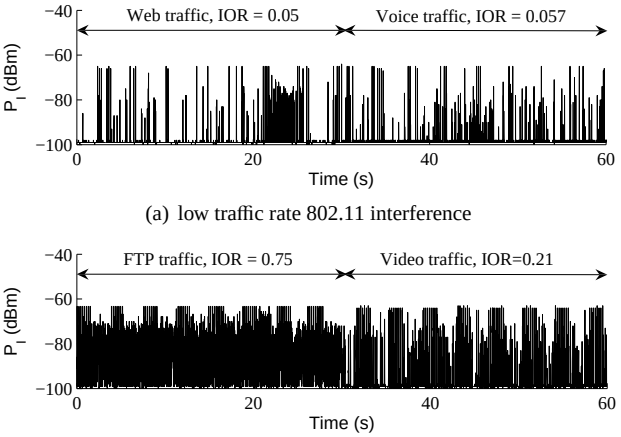


Fig. A· 1 SINR vs. PRR



(a) low traffic rate 802.11 interference

(b) high traffic rate 802.11 interference

Fig. A· 2 IOR values with different 802.11 applications

[10] CC2420, "2.4GHz IEEE 802.15.4 / zigbee RF transceiver", Chipcon, 2003.
[11] RT2800, "MIMO wireless chipset family", Ralink, 2007.
[12] TELOSB, "MOTE PLATFORM", Crossbow, 2005.
[13] I. Chakeres and L. Klein-Berndt, "AODVjr, AODV simplified," ACM SIGMOBILE Mobile Computing and Communication Review, pp. 100-101, Jul., 2002.

Appendix: Coordinate of base center of 3D cones

The Cartesian coordinates of base center of 3D cones defined by Platonic solids are as follows [9]:

- Tetrahedron:
  (1, 1, -1); (1, -1, 1); (-1, 1, 1); (-1, -1, -1)
- Hexahedron:
  (±1, 0, 0); (0, ±1, 0); (0, 0, ±1)
- Octahedron:
  (±1, ±1, ±1)
- Dodecahedron:
  (0, ±1, ±φ) (±1, ±φ, 0); (±φ, 0, ±1)
- Icosahedrons:
  (±1, ±1, ±1); (0, ±1/φ, ±φ); (±1/φ, ±φ, 0); (±φ, 0, ±1/φ)

where $\varphi = (1 + \sqrt{5})/2$ is the golden ratio.

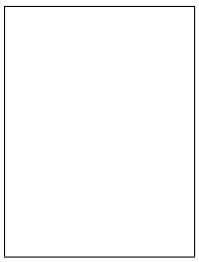
Appendix: Correlation between SINR and PRR

We present testbed experiment results to show the correlation between SINR and PRR. In experiments, 802.15.4 sender transmits 200 20-byte packets at each transmit power at the rate of ten packet per a second. 802.15.4 receiver measures the receive signal power and the interference power; and the computer connected with it calculates SINR value. We set the interference power as -88dBm (at the 802.15.4 receiver) by adjusting the transmit power of 802.11 nodes. Figure A· 1 shows that PRR has strong correlation with SINR regardless of the interference power. PRR is close to 100% when SINR is over a certain threshold, about 2dB.

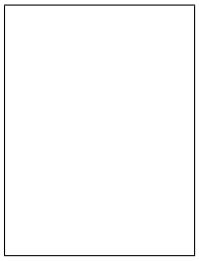
Appendix: 802.11 interference pattern

To understand the 802.11 interference pattern, we conduct field test with four 802.11 applications: web browser, VoIP messenger, FTP program, and video streaming player. In experiments, 802.11 nodes make the interference with different applications and a 802.15.4 node periodically estimates the interference power at the start of every wake-up period for a short time (it takes 128us). The interference occupy ratio (IOR) is defined as W_i/W where W_i and W are the number of wake-up periods detecting the interference and the total number of wake-up periods. When the channel is not heavily occupied as shown in Fig. A· 2(a), 802.15.4 nodes can succeed transmissions during the interval of 802.11 transmissions. When IOR $\geq$ 0.2 as shown in Fig. A· 2(b), the transmissions of 802.15.4 nodes are easily corrupted by the interference. Therefore, in testbed experiments of Section 5.2, each node checks IOR value every 50 wake-up periods and updates the signal power threshold if the IOR value is over 0.2.

Junseok Kim received the B.S degree and the M.S degree in Electronics Engineering from Konkuk University, Seoul, South Korea, in 2008 and 2010 respectively. He is now a doctorate student at the University of Arizona and a research member of Wireless and Advanced Networking Group under faculty advisor Marwan M. Krunz. He had published his first IEICE journal paper when he was a senior undergraduate student and wrote over 20 papers for various journals and conferences until now. His research interests include wireless ad-hoc (sensor) networks, vehicle ad-hoc networks, and cognitive-radio network.



Jongho Shin received the B.S degree in Civil Engineering from Korea University, Seoul, South Korea, in 1983 and the M.S degree in Civil and Environmental Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1982. He earned the Ph.D. degree from Department of Civil and Environmental Engineering, Imperial College of Science, Technology, and Medicine, London, England in 2000. He received a President Commendation in 2001 from South Korea Government and a John Henry Garrood King Medal in 2003 from Institution of Civil Engineers. From 2004 through the present, he is a professor in Konkuk University, Seoul, South Korea. He is now one of the senior secretaries to the President for economy policy in South Korea.



Younggoo Kwon received the B.S and M.S degrees in Electrical Engineering from Korea University, Seoul, South Korea, in 1993 and 1996, respectively. He earned the Ph.D. degree in Electrical Engineering from the Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA, in 2002. From 2002 to 2003, he was a Senior Member of the Research Staff at Samsung Electo-Mechanics Central R&D Center. Currently, he is an Associate Professor in the Department of Electronics Engineering, Konkuk University, Seoul, South Korea. He is a member of the IEICE.