Risk-Aware Beacon Scheduling for Tree-Based ZigBee/IEEE 802.15.4 Wireless Networks

Li-Hsing Yen*
Dept. Computer Science & Information Eng.
National University of Kaohsiung
Kaohsiung, Taiwan 811, R.O.C.
lhyen@nuk.edu.tw

Yee Wei Law Marimuthu Palaniswami†
Dept. Electrical & Electronic Eng.
The University of Melbourne
Parkville, Victoria 3052, Australia
{ywlaw,swami}@ee.unimelb.edu.au

ABSTRACT

In a tree-based ZigBee network, ZigBee routers (ZRs) must schedule their beacon transmission times to avoid beacon collisions. The beacon schedule determines packet delivery latency from the end devices to the ZigBee coordinator at the root of the tree. Traditionally, beacon schedules are chosen such that a ZR does not reuse the beacon slots already claimed by its neighbors, or the neighbors of its neighbors. We observe however that beacon slots can be reused judiciously, especially when the risk of beacon collision caused by such reuse is low. The advantage of such reuse is that packet delivery latency can be reduced. We formalize our observation by proposing a node pair classification scheme, that classifies pairs of nodes that are at most two hops apart. Based on this scheme, we can easily assess the risk of slot reuse by a node pair. If the risk is high, slot reuse is disallowed; otherwise, slot reuse is allowed. This forms the essence of our *ZigBee-compliant, distributed, risk-aware, probabilistic beacon scheduling algorithm*. Simulation results confirm that our algorithm produces a lower latency compared to if a more conventional slot reuse rule is used.

Categories and Subject Descriptors

COMPUTER-COMMUNICATION NETWORKS [**Network Architecture and Design**]: Wireless communication

General Terms

ZigBee/IEEE 802.15.4 beacon scheduling

*The first author is supported by the National Science Council, ROC, under grant NSC 97-2221-E-390-014.

†The second and third authors are supported by the Australian Research Council Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), and the DEST International Science and Linkage Grant.

WICON'08 November 17-19, 2008, Maui, Hawaii, USA.

1. INTRODUCTION

A wireless Personal Area Network (PAN), as defined by IEEE 802.15.4 [2], comprises devices that are characterized by low power, low data rate, short communication range, and low cost. IEEE 802.15.4 devices can be categorized into full function devices (FFDs) and reduced function devices (RFDs). FFDs are able to forward frames for other devices, while RFDs lack such capability. An FFD can initiate a PAN and act as the coordinator of the PAN. If beacon mode is enabled, the PAN coordinator defines a superframe that delineates the timing structure of the PAN. A superframe consists of active and inactive periods (Fig. 1). The active period begins with a beacon frame, by which nearby devices can identify the presence of the coordinator and thereby join the PAN by making *association*[1] with the coordinator. Beacon frames also serve the purpose of maintaining time synchronization between the coordinator and associated devices. Following the beacon is a number of time slots used for data exchange between the coordinator and associated clients. In the inactive period that follows the active period, no data traffic is expected in the PAN so devices can either enter power-saving mode or attempt communication with devices in other coexisting PANs.

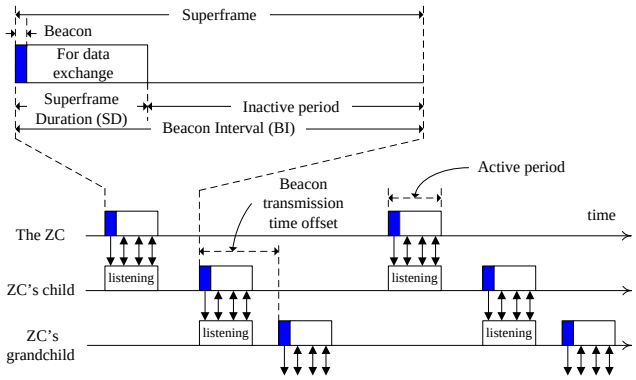

Figure 1: Example of a beacon schedule and the structure of a superframe.

1.1 Problem Statement

[1]Association is a communication primitive on the MAC sublayer, specified by the standard, by which a device associates itself with a particular PAN [6]. We say "associate" and "join" interchangeably.

The ZigBee specification [6] specifies how to organize a number of IEEE 802.15.4 devices into a star, tree, or mesh network topology. In a tree topology, RFDs join the tree as leaf nodes, referred to as ZigBee End Devices (ZEDs). The root is called the ZigBee Coordinator (ZC) and all internal nodes are called ZigBee Routers (ZRs). The ZC and the ZRs are functionally identical to PAN coordinators, and therefore can only be FFDs. Every ZC/ZR should periodically broadcast its own beacon and every ZR should track its parent's beacon. When a new ZR joins the network, it shall determine the time offset of its beacon transmission relative to its parent's. The determination should avoid *beacon collision*, which refers to the potential failure of receiving beacon/data frames by any device due to concurrent transmissions of beacon or data frames from multiple ZC/ZRs. The problem of *beacon scheduling* is to arrange the active period of each ZR to avoid potential beacon collisions. Fig. 1 shows an example of a beacon schedule. Since the ZC has the exclusive right to determine its beacon transmission time, only ZRs have to schedule their beacons to avoid potential collisions.

A straightforward approach to beacon scheduling is to avoid the overlapping of active periods. However, this may not be feasible when numerous ZRs are involved. It also increases packet delivery latency from each device to the ZC, as discussed later in this paper. A more practical strategy is to avoid potential beacon collisions while allowing overlapping active periods. For example, the specification [6] states that the active period of a ZR shall not overlap with that of any physical neighbor (i.e., any device within communication range) or of the parent of any physical neighbor. Prior work [3, 4] even disallows the overlapping of active periods between two ZC/ZRs that have some physical neighbor(s) in common. For example, ZRs 4, 5, and 6 in Fig. 2 are not allowed to overlap their active periods with one another, since doing so causes beacon collision problem at ZR 7.

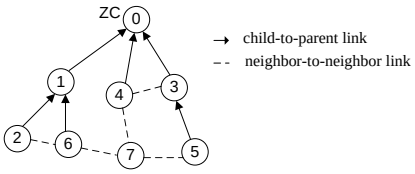

Figure 2: A ZigBee network with a tree topology.

We make two observations regarding the absolute preclusion of beacon collisions caused by the overlapping of active periods:

1. *A beacon schedule that initially appears collision-free may not appear to be collision-free to the devices joining the tree later.* Consider the example in Fig. 2. The nodes are numbered according to the order they join the network, which is configured in a tree topology. It is safe for ZR 6 to overlap its active period with that of ZR 4 or 5. Suppose that it overlaps its active period with ZR 5's. When ZR 7 joins the network later, ZR 7 is unable to associate with either ZR 5 or 6 due to the beacon collisions between these two ZRs. This potential conflict is inevitable due to the way by which ZigBee trees are constructed.

2. *The rule used to inhibit overlapping active periods may turn out to be too restrictive.* Using Fig. 2 as an example again, according to the rule, ZR 6 is not allowed to overlap its active period with ZR 2's because these two nodes are physical neighbors. However, as long as neither ZR 6 nor ZR 2 has a child in the future, no node will suffer from beacon collisions between ZR 6 and ZR 2.

Our conclusion from the above observations is that the absolute preclusion of beacon collisions caused by the overlapping of active periods is only possible when complete topology information of the tree is available. In reality, only partial topology information is available when a new ZR determines its active period. The rule used by the specification or prior work to inhibit overlapping of active periods also increases packet delivery latency from each device to the ZC, because the slots used by downstream nodes tend to be further apart from the slots used by upstream nodes, compared to the case where harmless slot reuse is tolerated. A more practical strategy is to estimate the *risk* of beacon collisions and then according to the risk decide whether or not to allow slot reuse.

1.2 Contribution and Organization

Our contribution is a *ZigBee-compliant, distributed, risk-aware, probabilistic beacon scheduling algorithm* that allows a node to locally assess the risk of slot reuse, and based on the assessed risk, to adopt the slot with the lowest latency to its parent. We express such a risk as *risk probability*, that is, the probability that slot reuse between two nodes will cause beacon collisions as seen by a future joining node. To assess this risk probability, we classify pairs of nodes that are at most two hops apart into Inhibited Pairs (IPs), Visible Pairs (VPs), Hidden Pairs (HPs) and Uninhibited Pairs (UPs); and according to the pair type, calculate the corresponding risk probability. The novelty of our work lies predominantly in this classification scheme and the estimation of this risk probability.

The rest of this paper is organized as follows. Section 2 covers the background of the problem. Section 3 discusses related work. In section 4, we describe our algorithm. Section 5 contains the simulation results that confirm the advantage of our algorithm. Finally, section 6 concludes.

2. PRELIMINARIES

The time between two consecutive beacons is called Beacon Interval (BI), while the duration of an active period is the Superframe Duration (SD). BI and SD in IEEE 802.15.4 are defined as follows.

$$\begin{cases} \text{BI} = aBaseSuperFrameDuration \times 2^{\text{BO}} \\ \text{SD} = aBaseSuperFrameDuration \times 2^{\text{SO}} \end{cases},$$

where $aBaseSuperFrameDuration = 15.36$ ms for a data rate of 250 kbps in the 2.4 GHz frequency band, and BO (Beacon Order) and SO (Superframe Order) are two integers ranging from 0 to 14. To ease beacon scheduling, we assume that all ZC/ZRs have identical BI setting and the whole BI is divided into non-overlapping time periods called *beacon slots*, each of which containing an *entire* active period of some ZC/ZR. It follows that the number of available beacon slots in a BI is $2^{\text{BO}-\text{SO}}$. This value is typically large to yield an energy-conserving low duty cycle (between $\sim 0.1\%$ and $\sim 2\%$ regardless of the frequency band [6]).

Table 1: Partial list of symbols

Symbol	Semantics
$2^{\text{BO-SO}}$	Number of beacon slots
Lm	Maximum depth of a tree
Cm	Maximum number of children of a ZC/ZR
Rm	Maximum number of children of a ZC/ZR that can be ZRs
$par(u)$	Parent of u
$pn(u)$	Physical neighbor of u
V	Vertex set representing all nodes in the network
E_t	Edge set defined by $\{(u,v)\|v = par(u)\}$
E_p	Edge set defined by $\{(u,v)\|v = pn(u), u = pn(v)\}$
G_t	Directed graph with vertex set V and edge set E_t
G_p	Undirected graph with vertex set V and edge set E_p
r	Radio range
R	Region where a WSN is deployed
A	Area of R
l	Width/height of R when R is square
p	Probability that a node is another node's physical neighbor
$J(u,v)$	Area of the region jointly covered by nodes u and v
$d(u,v)$	Distance between nodes u and v
$P(u,v)$	Expected probability that u's and v's slot sharing may prevent future neighbors of u and v to associate with either u or v
k	Node v's number of neighbors
γ,$\varphi(k)$	See Proposition 1

A tree-based ZigBee network is characterized by the parameters Lm, Cm and Rm (Table 1). According to the ZigBee specification, the ZC is at depth 0 and devices at depth Lm can only be ZEDs, not ZRs. Let $T(\text{Lm}, \text{Rm})$ be the maximal possible ZigBee tree (disregarding ZEDs) that can be formed, given Lm and Rm. The number of ZC/ZRs in $T(\text{Lm}, \text{Rm})$ is $\sum_{i=0}^{\text{Lm}-1} \text{Rm}^i = \frac{\text{Rm}^{\text{Lm}}-1}{\text{Rm}-1}$. Therefore, if

$$2^{\text{BO}-\text{SO}} \geq \frac{\text{Rm}^{\text{Lm}} - 1}{\text{Rm} - 1}, \tag{1}$$

then each ZC/ZR can be assigned a unique beacon slot for non-overlapping beacon transmissions. If (1) does not hold, which often happens in practice, some beacon slots must be reused to accommodate all possible ZRs in the network.

A ZED/ZR is allowed to send frames to its parent only during the parent's active period. If a ZR's active period is placed *later than* its parent's, all packets that it receives from its children during the current BI will not be forwarded to its parent until the next BI. Similarly, a ZR is unable to receive packets from its parent and then forward them to its children within the same BI, if the ZR's active period is placed *earlier than* its parent's. Therefore, uplink (from a device to its parent) and downlink (from a ZR to its child) packet delivery latencies are also determined by beacon scheduling. It is a challenge to schedule ZC/ZR's beacon transmissions to avoid potential collisions while minimizing packet delivery latency.

For the ensuing discussion, most of the symbols used are summarized in Table 1.

3. RELATED WORK

Koubâa et al. [3] have studied the problem of finding a collision-free beacon schedule with the assumption that each ZR can have its own independent BI/SD setting. A schedule was also provided when the given configuration is schedulable. However, this schedule is not latency-aware. Tseng and Pan [4] investigated the problem of finding a schedule that minimizes the maximal uplink packet delivery latency. They proved that this is an NP-hard problem and proposed two heuristic scheduling algorithms, namely centralized tree-based assignment and distributed slot assignment (DSA). Centralized tree-based assignment requires complete topology information as input and is not much better than DSA. In DSA, each node u chooses the slot that gives the lowest latency with respect to u's parent, but at the same time does not collide with the slots occupied by the nodes in u's $2r$-neighborhood.

All prior work [3, 4] emphasizes on reusing beacon slots whenever possible to minimize the number of needed slots. The feasibility of slot reuse depends not only on existing links but also on interfering neighbors. A node u is considered to interfere with another node v if u is within v's transmission range. For any node u in a ZigBee tree, let $par(u)$ denote u's parent in the tree. The tree can then be modeled as a directed graph $G_t(V_t, E_t)$, where V_t is the set of all ZigBee devices in the tree and $E_t = \{(u,v)|v = par(u)\}$. Let $G_p(V_t, E_p)$ be an undirected graph that models the underlying physical-layer topology of t, where $(u,v) \in E_p$ if devices u and v are within the transmission range of each other. The set of nodes in V_t that are adjacent to u in G_p are u's *physical neighbors*, denoted by $pn(u)$. All prior work avoids allocating u's beacon slot to another ZR v if $(u \in pn(v)) \vee (\exists w \in V_t - \{u,v\} : w \in pn(u) \wedge w \in pn(v))$. Take Fig. 2 as an example. The ZC and ZR 6 there are prohibited from taking the same beacon slot, because doing so would cause transmission collisions at ZR 1.

We argue that the rule imposed by prior work for slot reuse is however too restrictive. After all, reusing a beacon slot should be safe as long as there is no victim of such reuse. For any ZC/ZR, its beacon slot is for data exchange only between itself and its children. So when a ZC/ZR has no child, beacons from other ZC/ZRs in the vicinity cannot make victims at this ZC/ZR's site. For example, there is no victim when ZRs 1 and 3 in Fig. 2 take the same beacon slot, although they have a physical neighbor in common (i.e., the ZC). This kind of slot-reuse is permitted by the ZigBee specification. However, the specification does not allow slot reuse between physical neighbors, though such a reuse does not necessarily make a victim. The ZR pair $(2, 6)$ in Fig. 2 is an example.

In the next section, we will present a systematic approach for identifying the cases when slot reuse is relatively safe. This is done by classifying pairs of nodes that are at most two hops from each other into different *pair types*, and calculating the associated *risk probability*. High risk probability means the corresponding node pair would at a high probability prevent a future neighboring node from joining, and the node pair should therefore not share a beacon slot. Conversely, low risk probability means it is relatively safe for the node pair to share the same beacon slot.

4. PROBABILISTIC RISK-AWARE BEACON SCHEDULING

In this section, we first describe our node pair classification scheme, then our risk probability assessment methodology, and finally our ZigBee-compliant, distributed, risk-aware, probabilistic beacon scheduling algorithm. The node

pair classification scheme and risk probability assessment methodology allows the algorithm to reduce latency by reusing beacon slots, while still minimizing the risk of new nodes not being able to join the tree.

4.1 Node Pair Classification

We classify a ZC/ZR pair (u, v) as one of the following types based on a given G_t and a given G_p (Fig. 3):

- **Inhibited Pair** (IP): u and v are physical neighbors, and either u or v has a child; or, u and v are not physical neighbors, but u and v have a common neighbor which is a child of either u or v.
- **Visible Pair** (VP): u and v are physical neighbors but neither u nor v has any child.
- **Hidden Pair** (HP): u and v are not physical neighbors but have physical neighbors in common, although all these physical neighbors are neither u's nor v's children.
- **Unrelated Pair** (UP): u and v are not physical neighbors, neither do they have physical neighbors in common.

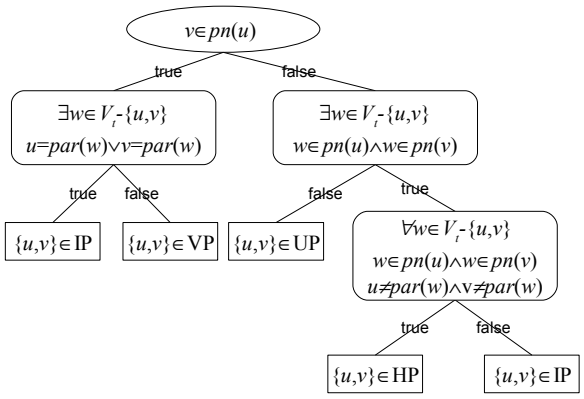

Figure 3: Proposed classification tree for identifying types of a ZC/ZR pair (u, v).

Fig. 4 shows some examples of VP and HP pairs that are safe to take the same beacon slot.

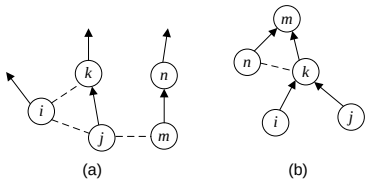

Figure 4: Examples of ZC/ZR pairs that are safe to take the same beacon slot: (a) VP $= \{(i, j), (j, m)\}$ (b) HP $= \{(i, j), (i, n)\}$.

The difference among various slot-reuse rules, including ours, is summarized in Table 2. From this table we can see our rule reuses beacon slots to the furthest extent.

Regardless of the slot-reuse rule, the complete G_t and G_p must be available to make a slot-reuse schedule *absolutely* collision-free. This requirement is considered impractical due to the way by which ZigBee trees are constructed. A device must hear beacon frames from a ZC/ZR before it can send a join request to the ZC/ZR. Before receiving the request, the ZC/ZR is not aware of the presence of that device. This means that ZC/ZR's beacon transmission schedule is determined with only partial knowledge of the network topology. Therefore, a ZigBee-compliant beacon scheduling method should be an on-line algorithm by nature: *slot numbers are assigned while tree construction is in progress.* Our algorithm is ZigBee-compliant because it respects exactly this principle.

Table 2: Comparison of slot-reuse rules

Beacon slot reusable?	IP	VP	HP	UP
Prior work [3, 4]	No	No	No	Yes
The specification [6]	No	No	Yes	Yes
Our rule	No	Yes	Yes	Yes

An instance of slot reuse is *risky* if it is safe at the time of its creation but may cause beacon collisions to some devices (ZRs or ZEDs) joining the network later. Using our new pair type terminology, this means that $(u, v) \in \text{VP} \cup \text{HP} \cup \text{UP}$ at time t, and some device w joining the network at a later time $t' > t$ may not be able to detect and thereby associate with u and v due to beacon collisions. Note that the instance of reuse does not endanger any already established association at time t. Slot reuse is risky only in the sense that it decreases the number of ZC/ZRs which w can associate with at t'. Node w may still find and associate with another ZC/ZR, but there is a non-zero probability that w fails to join the tree due to the absence of other associable ZC/ZRs. We assume that all collided beacons and data frames are garbled. Therefore, w can only associate with a ZC/ZR that does not cause beacon collisions at w. Once the association succeeds, it cannot be affected by the slot reuse between u and v.

In summary, as opposed to existing proposals, we propose reusing beacon slots by VPs, HPs and UPs when the risk is low. Given any *online* beacon scheduling algorithm that uses only local topological information, there is no guarantee that a good schedule would not become a bad schedule when a new node joins the network later, but our approach is to minimize such probability.

4.2 Assessment of Slot Reuse Risk

A key task in our algorithm is to determine the risk probability of slot reuse for a given device pair (u, v), based on current knowledge of G_p and G_t. We assume that the deployment of ZigBee devices is random yet follows a uniform distribution over a region R. Let A be the size (area) of R. Each device is assumed to have a radio communications range of r. A piece of area in R is said to be covered by a device u and termed u's coverage if every point in this area is within the communications range of u. We use $J(u, v)$ to denote the region jointly covered by two devices u and v, and use $d(u, v)$ to denote the physical distance between them.

LEMMA 1. *The expected area of the region jointly covered*

by two nodes u and v is

$$E[J(u,v)] = \begin{cases} \left(\pi - \frac{3\sqrt{3}}{4}\right) r^2 & \text{if } d(u,v) \le r, \\ \frac{\sqrt{3}}{4} r^2 & \text{if } r < d(u,v) \le 2r. \end{cases}$$

PROOF. See Appendix. □

A node placed near the boundary of R will cover less area than expected, as part of its coverage is outside R. This is referred to as the border effect. To avoid clumsy results caused by the border effect, the following analysis assumes that the region covered by any node is completely within R. If R is a rectangle area, the assumption can be achieved by adopting the torus convention [1], which turns a flat rectangle into a torus. With this assumption, the probability of link occurrence [5] is $p = \pi r^2/A$. Our core result is Proposition 1.

PROPOSITION 1. *Assume nodes are uniformly distributed in a region R of size A, and $A \gg \pi r^2$. Assume also all collided beacons and data grames are garbled. Suppose when a new node w joins, a pair of nodes u and v are using the same beacon slot. Denote by k the number of neighbors v has, and by $P(u,v)$ the expected probability that w can neither associate with u nor with v, then*
if $(u,v) \in$VP, $P(u,v) = P_V(u,v) = \left(1 + \frac{3\sqrt{3}}{4\pi}\right) p$;
if $(u,v) \in$HP, $P(u,v) = P_H(u,v) \approx \gamma p$;
if $(u,v) \in$UP, $P(u,v) = P_U(u,v) \approx \left[\gamma + \left(\frac{\sqrt{3}}{4\pi} - \gamma\right) \varphi(k)^{-1}\right] p$;
where $p = \frac{\pi r^2}{A}$, $\gamma \approx 0.17$, $\varphi(k) = \frac{2}{3} \int_{\theta=0}^{2\pi/3} \left[1 - \frac{\theta - \sin\theta}{\pi}\right]^k \sin\theta d\theta$.
Note: If $(u,v) \in IP$, then the common neighbor of u and v is already a victim of beacon collisions between u and v.

PROOF. We first look at the case $(u,v) \in$ VP. By the definition of VP, u and v are physical neighbors but neither u nor v has any child. $P_V(u,v)$ is the expected probability of w appearing in the neighborhood of either u or v. Based on the assumption that the nodes are uniformly distributed, and the inclusion-exclusion principle, the expected probability is $(2\pi r^2 - E[J(u,v)])/A$. Now, invoking Lemma 1 with the condition $d(u,v) \le r$, we have

$$P_V(u,v) = \frac{2\pi r^2 - E[J(u,v)]}{A} = \left(1 + \frac{3\sqrt{3}}{4\pi}\right) p.$$

We now look at the cases $(u,v) \in$ HP and $(u,v) \in$ UP together, because they are closely related. Let us define the following events which we will use later:

W: the new node w becomes a neighbor of u and v;

X: $r < d(u,v) \le 2r$;

Y: u and v do not share any neighbor;

ζ_i: v's ith neighbor is a child of either u or v.

By the definition of HP, u and v are not physical neighbors but have physical neighbors in common, although these physical neighbors are neither u's nor v's children. $P_H(u,v)$ is the expected probability of w becoming u's and v's common neighbor, which is

$$\Pr[W|\bar{Y} \wedge \bigcap_{i=1}^{k} \bar{\zeta}_i, X],$$

but since the event W is independent of ζ_i,

$$P_H(u,v) = \Pr[W|\bar{Y} \wedge \bigcap_{i=1}^{k} \bar{\zeta}_i, X] = \Pr[W|\bar{Y}, X]. \quad (2)$$

By the definition of UP, u and v are not physical neighbors and have no common neighbors. $P_U(u,v)$ is the expected probability of w becoming a new common neighbor of u and v. Using the same argument as before,

$$P_U(u,v) = \Pr[W|Y, X]. \quad (3)$$

It is easy to see that

$$\Pr[W|X] = \Pr[W|Y,X] \Pr[Y|X] + \Pr[W|\bar{Y},X] \Pr[\bar{Y}|X] \quad (4)$$

In (4), we can find $\Pr[W|X]$ by simply invoking Lemma 1 with the condition $r < d(u,v) \le 2r$

$$\Pr[W|X] = \frac{E[J(u,v)]}{A} = \frac{\sqrt{3}}{4\pi} p. \quad (5)$$

Also, in (4), we can find $\Pr[Y|X]$ as follows. If we denote the area of the hatched region in Fig. 5 as $J(u,v)$, and if v has k neighbors, the probability of all v's k neighbor not falling in the hatched region is $[1 - J(u,v)/(\pi r^2)]^k$. From the Appendix, $J(u,v) = r^2(\theta - sin(\theta))$, where $0 \le \theta \le 2\pi/3$ is defined as shown in Fig. 5. To get the expectation of $[1 - J(u,v)/(\pi r^2)]^k$, we need to integrate the expression over $[0, 2\pi/3]$:

$$\begin{aligned} \Pr[Y|X] &= E[\{1 - r^2(\theta - \sin\theta)/(\pi r^2)\}^k] \\ &= \int_{\theta=0}^{2\pi/3} \left[1 - \frac{\theta - \sin\theta}{\pi}\right]^k g(\theta) d\theta, \end{aligned}$$

where $g(\theta)$ is the probability density function (p.d.f.) of θ. From the Appendix, $g(\theta) = \frac{2}{3}\sin\theta$, hence

$$\Pr[Y|X] = \frac{2}{3} \int_{\theta=0}^{2\pi/3} \left[1 - \frac{\theta - \sin\theta}{\pi}\right]^k \sin\theta d\theta, \quad (6)$$

which is solely a function of k. For simplicity, we denote the RHS of (6) as $\varphi(k)$, i.e., $\Pr[Y|X] = \varphi(k)$. Fig. 6 shows that $\varphi(k)$ is very close to simulation results.

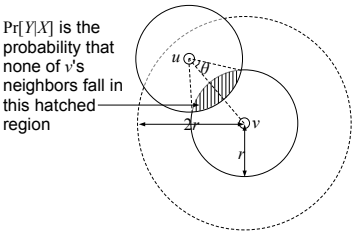

Figure 5: Deriving $\Pr[Y|X]$.

Subsituting (5) and (6) back into (4), we have

$$\Pr[W|Y,X] = [1 - \varphi(k)^{-1}] \Pr[W|\bar{Y},X] + \frac{\sqrt{3}}{4\pi} \varphi(k)^{-1} p. \quad (7)$$

Using (7), if we can find $\Pr[W|\bar{Y},X]$, then we can find $\Pr[W|Y,X]$. But finding $\Pr[W|\bar{Y},X]$ analytically is non-trivial. We resort to an experimental approach. From our experiments (detail in Appendix), we found that $\Pr[W|\bar{Y},X]$ is approximately linear with p, and if we denote the linear constant by γ, then

$$\gamma \approx 0.17. \quad (8)$$

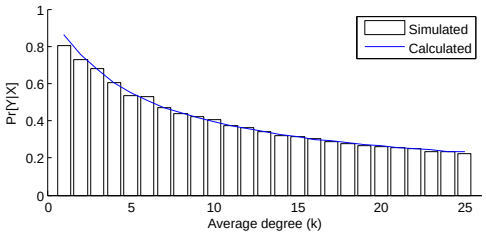

Figure 6: Comparison of (6) to simulation results.

Substituting (8) in (7), we get

$$\Pr[W|Y,X] \approx \left[\gamma + \left(\frac{\sqrt{3}}{4\pi} - \gamma\right)\varphi(k)^{-1}\right] p. \qquad (9)$$

Finally, substituting (8) into (2), and (9) into (3) respectively, we have

$$P_H(u,v) \approx \gamma p,$$
$$P_U(u,v) \approx \left[\gamma + \left(\frac{\sqrt{3}}{4\pi} - \gamma\right)\varphi(k)^{-1}\right] p.$$

□

By just comparing the coefficients of p in Proposition 1, we can observe that slot reuse between a VP has the highest risk, while that between an UP has the lowest. Fig. 7 shows the risk probabilities of slot reuse for VP, HP, and UP with respect to link probability p for the case $n = 300$. Notice P_V is for most cases much higher than P_H and P_U, and rises much faster than the latter two. In other words, when the radio range is much smaller than the deployment area, the risk of slot reuse by VPs is small, whereas the risk of slot reuse by HPs and UPs is negligible; but as the radio range becomes bigger with respect to the deployment area, the risk of slot reuse by VPs becomes bigger and is the most dominant risk.

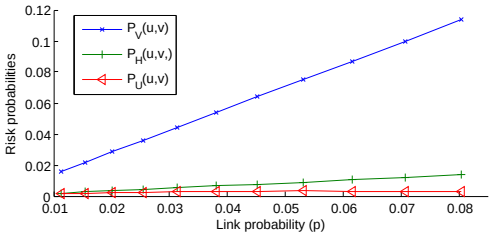

Figure 7: Risk probabilities of slot reuse when $n = 300$.

4.3 The Algorithm

We first describe our core algorithm, and then variations of the core algorithm. Our description starts with what happens right after the sensor nodes are deployed. While the sink claims beacon slot 0, the other sensor nodes go into sleep mode for a random amount of time. When a node u wakes up, it tries to find a parent to associate with. If it finds a suitable parent candidate, it runs the following procedure to pick a slot:

Let i_p be u's candidate parent's slot
$i \leftarrow (i_p - 1) \mod k$
while slot i is already used by a neighboring ZR v and there are still slots to try
do
  if $(u,v) \in$ rejected pair type **then**
    $i \leftarrow (i-1) \mod k$
  else
    u calculates $P(u,v)$
    u accepts slot i at a probability of $1 - P(u,v)$
    if u accepts slot i
      return i as u's slot number
    else
      $i \leftarrow (i-1) \mod k$
    end if
  end if
end while
return failure

By "accepts slot i at a probability of $1 - P(u,v)$", we mean the device generates a pseudorandom number in the range [0, MAX], and if the number falls in the range [0, $P(u,v)$·MAX], the device rejects slot i; otherwise the device accepts slot i. Notice that for efficient convergecasts, a child picks a smaller slot number than its parent's, this is the reason why we decrement i for each try.

If the node fails to pick a slot, it announces itself as a ZED and the slot it follows (i.e., the slot its parent claims). Otherwise, it announces itself as a ZR and the slot it claims. From time to time, a node will discover a better parent candidate (e.g., some node that is closer to the sink than its current parent), and when it does so, it will disassociate with its current parent and associate with the new parent candidate. From time to time also, a node might discover beacon collision between its parent and one of its ZR neighbor, and when this happens, it will also disassociate with its current parent and associate with another ZR neighbor whose beacons are not corrupted (recall only ZRs can be parents). All in all, a node only needs to know the slots its neighbors are occupying/following. Therefore, this algorithm is totally distributed. However, in the core algorithm, a node can only discover the slots occupied or followed by its neighbors, hence a node can only discover IPs and VPs, and not HPs or UPs.

We can extend the core algorithm by considering whether to make the algorithm distributed or centralized, or controlling the extent of information a ZR/ZED broadcasts. The simplest variations are listed in Table 3 and are described in detail as follows:

- **Distributed algorithms that detect only IPs and VPs**: These algorithms can only detect IPs and VPs because every node only broadcasts about the slot it occupies/follows, and so a node hears only about the slots its neighbors occupy/follow. The variants are algorithms D1VHU and D1HU. In fact, the core algorithm we described earlier is exactly D1VHU. D1HU is different from D1VHU in that it rejects slot reuse by VPs, and in this sense it implements the slot reuse strategy proposed by the ZigBee standard. All algorithms in this class accept slot reuse by HPs and UPs implicitly.

- **Distributed algorithms that detect IPs, VPs, HPs but not UPs**: These algorithms can detect HPs in addition to IPs and VPs because every node not only broadcasts about the slot it occupies/follows, but

also the slots its neighbors occupy/follow. The variants are algorithms D2VHU, D2HU and D2U. D2VHU and D2HU are extension of D1VHU and D1HU respectively, with the benefit of the network reaching steady state ealier. D2U rejects both VPs and HPs, and in this sense it implements the slot reuse strategy proposed by prior work [3, 4] in the literature. All algorithms in this class accept slot reuse by UPs implicitly. UPs by definition cannot be detected by any local/distributed algorithms.

- **Centralized algorithms that detect all IPs, VPs, HPs and UPs**: With complete topology information, a ZC can detect all pair types. The variants are algorithms CVHU, CHU, CU and C. From a practical viewpoint, these algorithms are much less scalable and much less energy-efficient than the distributed variants. We only mention them here for comparison with the distributed variants later.

In summary, our core algorithm is ZigBee-compliant because it can be executed while the ZigBee itself is being formed. It is distributed because only local information is required. It is risk-aware, because it can calculate the risk of slot reuse by using Proposition 1. It is probabilistic, because it accepts/rejects slot reuse at a probability exactly dictated by the calculated risk.

In the next section, simulation will show that in terms of packet delivery latency, (1) our core algorithm is better than Tseng et al.'s DSA, and (2) that the slot reuse rule represented by our core algorithm is better than the slot reuse rules espoused by the specification and prior work in the literature.

Table 3: Variations of the core algorithm D1VHU

	Distributed 1		Distributed 2			Centralized			
VP	✓	×	✓	×	×	✓	×	×	×
HP	•	•	✓	✓	×	✓	✓	×	×
UP	•	•	•	•	•	✓	✓	✓	×
Name	D1VHU	D1HU	D2VHU	D2HU	D2U	CVHU	CHU	CU	C

Legend:

- ✓ accepts slot reuse at probability $1 - P(u, v)$
- × rejects slot reuse
- • accepts implicitly

The algorithms are labeled according to whether they are distributed or centralized, and what pair type they accept for slot reuse. For example, the distributed algorithm that detects only IPs and VPs, and accepts slot reuse for VPs (probabilistically), HPs (implicitly) and UPs (implicitly) is labeled D1VHU. Note that all algorithms reject slot reuse by IPs.

5. SIMULATION RESULTS

We have two objectives for our simulations: (1) to compare our algorithms with Tseng et al.'s DSA in terms of average maximum latency and average latency; and (2) to compare the algorithms in Table 3 in terms of average latency and number of associated nodes – the latter comparison is interesting because it tells us if the slot reuse rule represented by our core algorithm is better than the slot reuse rules espoused by the specification and prior work in the literature.

To compare with Tseng et al.'s DSA, we use Tseng et al.'s simulation settings and metrics. The simulation settings are summarized on top of each sub-figure in Fig. 8. The metrics used by Tseng et al. is average maximum latency, i.e., if for the ith simulation run, the maximum latency is L_i, then the average maximum latency is $\sum_i L_i/max(i)$. Additionally, we also use the more conventional metric of average latency, which is defined as the average of the average latencies of all simulation runs. As evident in Fig. 8, there are five performance categories:

1. D1VHU, D2VHU and CVHU perform similarly and are the best performers. The fact that CVHU perform similarly to the other distributed algorithms mean that there is no advantage in being able to detect all pair types. Therefore in practice, either D1VHU or D2VHU is recommended with the following trade-off in mind: D1VHU requires less transmission by each ZR/ZED but require more time to reach steady state, whereas D2VHU has the opposite requirement.

2. D1HU, D2HU and CHU perform similarly and are quite worse than the previous category. This observation confirms that allowing slot reuse by VPs judiciously, as the previous category does, reduces latency.

3. D2U and CU perform similarly and are much worse than the previous category. This observation confirms that allowing slot reuse by HPs judiciously, as the previous category does, reduces latency.

4. Tseng et al.'s DSA is quite worse than the previous category. DSA does not allow slot reuse between nodes at most $2r$ apart. This observation confirms that allowing slot reuse by UPs judiciously, as the previous category does, reduces latency. Our numbers for DSA are consistent with the numbers published by Tseng et al. in their original paper [4].

5. Algorithm C is the worst performer since it does not allow any slot reuse at all. This serves as the benchmark for the worst performers in beacon scheduling algorithms.

We also see that both metrics average maximum latency and average latency give the same comparison. In fact, if not to compare with Tseng et al.'s DSA, average latency should be preferred because it gives lower standard deviations.

In the following simulations, we compare D2VHU, D2HU, D2U and C (each representing its own class) with Tseng et al.'s DSA in terms of average latency and number of associated devices. Fig. 9(a)(b)(c) show that regardless of the 'shape' of the tree (the 'shape' is adjusted by configuring the ZigBee parameters Lm, Rm and Cm), D2VHU, D2HU and D2U consistently outperform DSA. Algorithm C remains much worse than the rest.

In terms of number of associated devices, DSA can admit the most devices in its tree. Due its aggressive slot-reuse policy, D2VHU can accommodate the least number of devices, because for example, when a VP uses the same beacon slot, the pair can no longer admit any child. The difference between D2VHU and DSA is marginal for 'flat' trees (Fig. 9(d)). The difference becomes larger as the tree becomes 'taller' (Fig. 9(e)(f)). Note though if we do not restrict Lm, D2VHU can admit virtually as many devices as

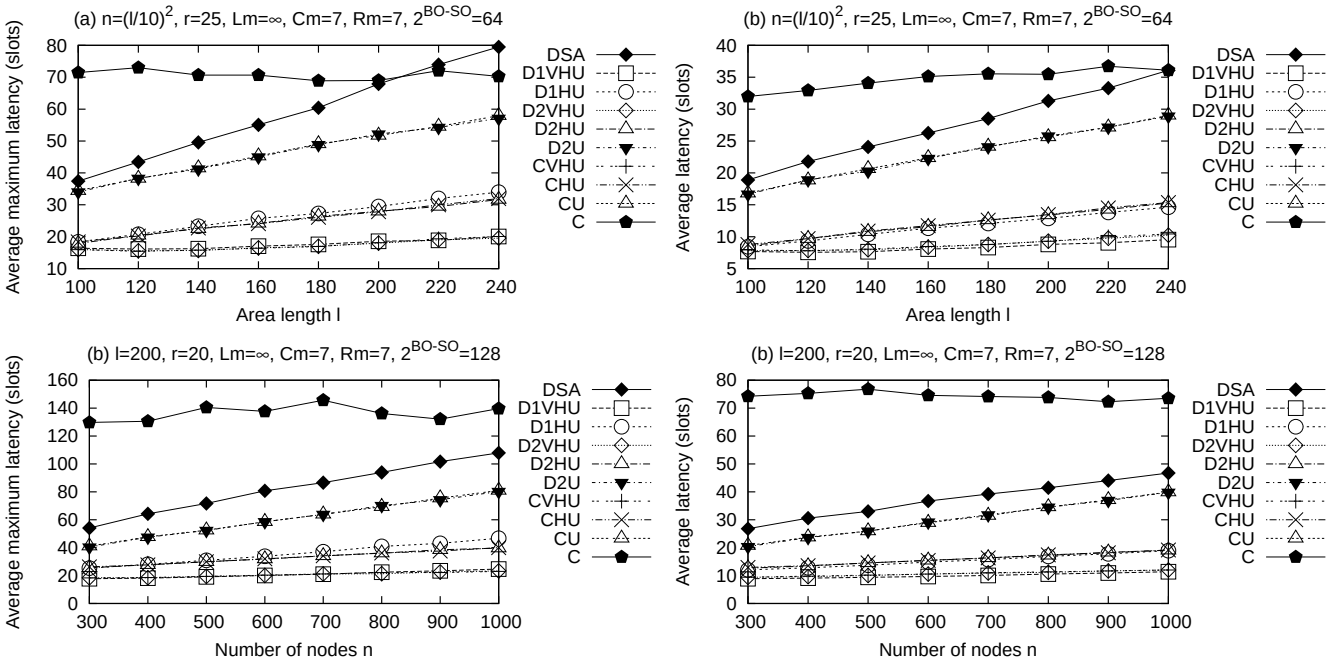

Figure 8: Comparing our algorithms with Tseng et al.'s DSA in terms of (a) average maximum latency, and (b) average latency, by fixing network density; comparing our algorithms with Tseng et al.'s DSA in terms of (c) average maximum latency, and (d) average latency, by varying network density.

DSA can – no graph is shown for this case though, because the resultant graph would show two almost coinciding lines.

6. CONCLUSIONS AND FUTURE WORK

Traditionally, beacon schedules are chosen such that a ZR does not reuse beacon slots already claimed by its neighbors, or the neighbors of its neighbors. We observe however that beacon slots can be reused judiciously to the desirable effect of reduced packet delivery latency. Based on this idea, we have formalized a framework where we can calculate the risk of slot reuse between two nodes that are at most two hops apart. If the calculated risk is high, slot reuse is disallowed; otherwise, slot reuse is allowed. This is essentially the heart of our *ZigBee-compliant, distributed, risk-aware, probabilistic beacon scheduling algorithm*. Simulation results confirm that (1) our core algorithm is better than Tseng et al.'s DSA, and (2) the slot reuse rule represented by our core algorithm is better than the slot reuse rules espoused by the specification and prior work in the literature.

Future work includes foremost a more rigorous derivation of $P_H(u,v)$ and $P_U(u,v)$. More extensive simulations are also required to estimate the time required by D1VHU and D2VHU to reach steady state. Due to the scale of a ZigBee implementation and the number of configurable parameters, we defer the detailed comparison to a future paper.

7. REFERENCES

[1] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley and Sons, 1988.

[2] IEEE. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), Oct. 2003.

[3] A. Koubâa, M. Alves, M. Attia, and A. Nieuwenhuyse. Collision-free beacon scheduling mechanisms for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks. In *7th Int'l Workshop on Applications and Services in Wireless Networks, Santander, Spain*, May 2007.

[4] Y.-C. Tseng and M.-S. Pan. Quick convergecast in ZigBee/IEEE 802.15.4 tree-based wireless sensor networks. In *ACM Int'l Workshop on Mobility Management and Wireless Access, Terromolinos, Spain*, Oct. 2006.

[5] L.-H. Yen and Y.-M. Cheng. Clustering coefficient of wireless ad hoc networks and the quantity of hidden terminals. *IEEE Commun. Lett.*, 9(3):234–236, Mar. 2005.

[6] ZigBee Standards Organization. *ZigBee Specification*, Dec. 2006.

APPENDIX

Proof of Lemma 1

For the case $d(u,v) \leq r$, it is known [5] that the expected size of $J(u,v)$ is $\left(\pi - \frac{3\sqrt{3}}{4}\right) r^2$.

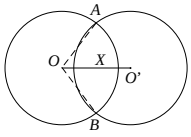

Figure 10: Two disks intersecting each other.

We now investigate the case $r < d(u,v) \leq 2r$. Suppose u and v are located at O and O', respectively. Let $X = d(u,v)$

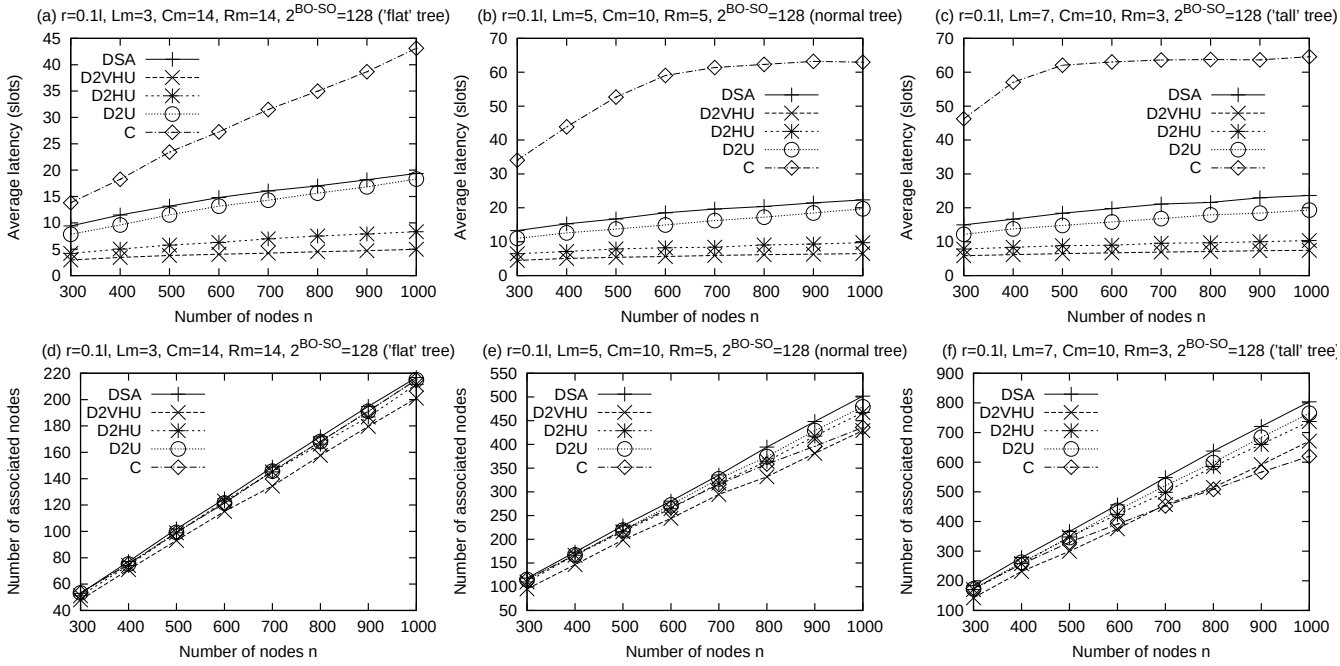

Figure 9: Comparing algorithms in terms of average latency for (a)'flat' networks, (b) normal networks and (c) 'tall' networks; comparing algorithms in terms of number of associated devices for (d) 'flat' networks, (e) normal networks and (f) 'tall' networks.

be a random variable with range $r < X \leq 2r$. We want to calculate the expected size of the area jointly covered by u and v, which is a lens-shaped region. Let A and B be two end points of the lens (refer to Fig. 10). The area of each half of the lens is equal to the area of sector OAB minus the area of triangle OAB. Let $\theta = \angle AOB$ be the central angle given X, where $0 \leq \theta < 2\pi/3$. We have $X = 2r\cos(\theta/2)$. The area of triangle OAB is

$$\left[\frac{2r\sin\left(\frac{\theta}{2}\right)\frac{X}{2}}{2}\right] = r^2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) = \frac{r^2\sin\theta}{2}.$$

So the area of the lens is

$$2\left[\frac{\pi r^2\theta}{2\pi} - \frac{r^2\sin\theta}{2}\right] = r^2(\theta - \sin\theta).$$

Let $F(x)$ be the probability distribution function (p.d.f.) of X. Since nodes are uniformly distributed, $\Pr[X \leq x]$ is proportional to the size of the ring-shaped region centered at O with inner and outer radii r and x, respectively. Therefore,

$$F(x) = \Pr[r < X \leq x] = \frac{\pi x^2 - \pi r^2}{4\pi r^2 - \pi r^2} = \frac{x^2 - r^2}{3r^2}.$$

Since $\theta = 2\arccos(X/2r)$, the p.d.f. of θ is

$$\begin{aligned} G(y) &= \Pr\left[0 \leq \theta \leq y\right] = \Pr\left[2r\cos\frac{y}{2} \leq X \leq r\right] \\ &= F(r) - F\left(2r\cos\frac{y}{2}\right) = -\frac{2}{3}\cos y - \frac{1}{3}. \end{aligned}$$

It follows that the probability density function of θ is $g(y) = G'(y) = \frac{2}{3}\sin y$. Therefore, the expected area of the lens-shaped region that is jointly covered by O and O' is

$$\int_0^{\frac{2\pi}{3}} r^2(\theta - \sin\theta)(\frac{2}{3}\sin\theta)d\theta = \frac{\sqrt{3}r^2}{4}.$$

Estimating $\Pr[W|\bar{Y}, X]$

By plotting $\Pr[W|\bar{Y}, X]$ against p (Fig. 11), we can see that $\Pr[W|\bar{Y}, X]$ is approximately a linear function of p. By robust-fitting the function $\Pr[W|\bar{Y}, X] = \gamma p$ to the plots, we find the estimated values of γ and their corresponding coefficients of determination (R^2) in Table 4. By deriving a weighted average of the estimated values of γ in Table 4, i.e., $E[\Gamma] = \frac{\sum_i \gamma_i R_i^2}{\sum_i R_i^2}$, we get $\gamma \approx 0.17$.

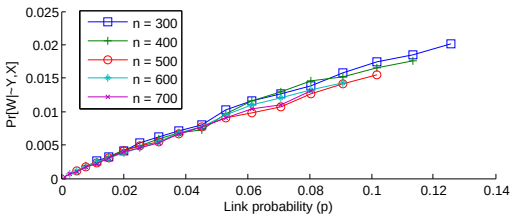

Figure 11: $\Pr[W|\bar{Y}, X]$ against p, with 15 data points per value of n

Table 4: Estimated values of γ vs n

n	300	400	500	600	700
γ	0.1761	0.1748	0.1669	0.1912	0.1781
R^2	0.9726	0.9834	0.9819	0.9617	0.9974