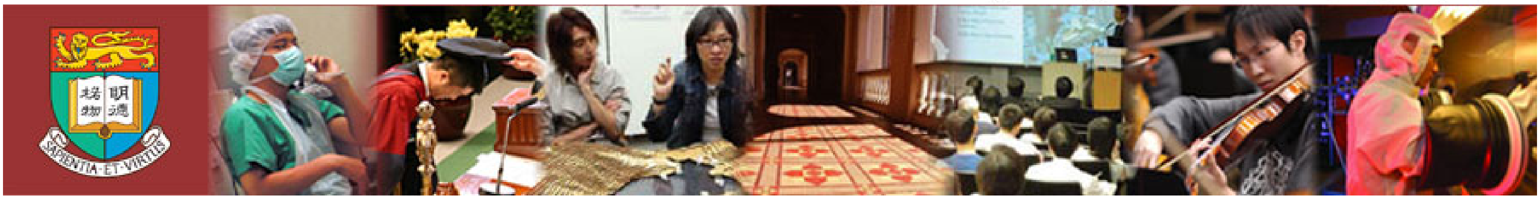


Is Topology-Transparent Scheduling Really Inefficient in Static Multihop Networks?

Yiming Liu, Victor O. K. Li, *Fellow, IEEE,* Ka-Cheong Leung, *Member, IEEE,* and Lin Zhang, *Member, IEEE*

***Abstract*—Topology-transparent scheduling algorithms are oblivious to the network topology changes and can provide throughput and delay guarantees in mobile multihop networks. However, it has been argued that topology-transparent scheduling algorithms are inefficient when the network is static, compared to topology-dependent scheduling algorithms. In this paper, we propose to utilize both assigned and unassigned slots efficiently to boost the performance of topology-transparent scheduling algorithms. We conclude that, in certain cases, the performance of the proposed topology-transparent scheduling algorithm can be comparable to or better than that of some topology-dependent algorithms even when the network topology remains unchanged. Yet, the proposed algorithm also works even when the network topology is dynamic.**

***Index Terms*—Topology-transparent scheduling, topology-dependent scheduling, efficiency.**

I. Introduction

TRANSMISSION scheduling plays a critical role in wireless networks. Many medium access control (MAC) protocols have been developed for ad hoc networks. The contention-based approaches, such as Carrier Sense Multiple Access (CSMA), cannot provide deterministic delay and throughput bounds. It is also shown that contention-based approaches suffer from serious instability and unfairness issues in multi-hop ad hoc networks [9]. Previous approaches in topology-dependent scheduling [3, 7, 8] require each node to maintain accurate network topology information. Information exchanges and re-computation are required to maintain accurate network topology information and to distribute the new schedules when the network topology changes. Thus, the robustness and effectiveness of these algorithms are undermined in highly dynamic wireless networks. To overcome the aforementioned disadvantages, topology-transparent scheduling algorithms have been proposed [1, 2, 4, 5].

It has been shown that topology-transparent scheduling algorithms work well in mobile multihop networks. They are oblivious to the network topology changes and can provide throughput and delay guarantees. However, it is always argued that topology-transparent scheduling algorithms are not efficient in static networks, compared to topology-dependent scheduling algorithms. In this letter, we propose to improve the throughput of topology-transparent scheduling algorithms by utilizing the unassigned slots in a collision-free manner. We show that, in certain cases, the proposed topology-transparent algorithm may outperform some topology-dependent algorithms even when the network topology is static.

II. System Model

A multihop network can be represented by a graph $G(V,E)$. V is the set of all network nodes and E is the set of all edges. If Node v is within the interference range of Node u, an edge denoted by (u,v) is in E. We assume that if $(u,v) \in E$, $(v,u) \in E$. To simplify the calculation, we assume that the interference range of a node equals its transmission range. Let $N_1(u)$ and $N_2(u)$ denote the sets of one-hop neighbors and two-hop neighbors of Node u, respectively. Thus, $N_1(u) \subset N_2(u)$. The degree of a node u, $D(u)$, is defined as the number of one-hop neighbors of Node u, i.e., $D(u) = |N_1(u)|$. The maximum degree $D_{\max}$ is defined as $D_{\max} = \max\limits_{u \in V} D(u)$. We assume that $D_{\max}$ is much smaller than the number of nodes N and remains constant in such multihop networks [3].

We focus on TDMA networks. Time is divided into equal-sized frames. Each frame is assigned an index t, where $t = 0, 1, \ldots$, and further divided into three parts. The first two parts are intended for control packets and called control frames (CF1 and CF2). The last part is meant for data packets and called data frame (DF), as illustrated in Fig. 1. Each of the control and data frames is divided into q subframes, each of which consists of p synchronized slots. We use the optimal frame structure that achieves the maximum guaranteed throughput in [5]. That is, we set $p = q$, which equals the smallest prime or prime power that satisfies $p = q \geq 2kD_{\max}$ if $N^{k+1} \leq 2kD_{\max}$, and the smallest prime or prime power that satisfies $p = q \geq N^{k+1}$ otherwise. It is also proved that $k = 1$ for most cases [5]. Thus, without loss of generality, we use $k = 1$ in the following for simplicity. A mini-slot for the acknowledgment is piggybacked at the end of each data slot. Synchronization can be achieved by Global Positioning System (GPS). In practice, the length of DF should be much larger than that of CF1 and CF2 to reduce the overhead.

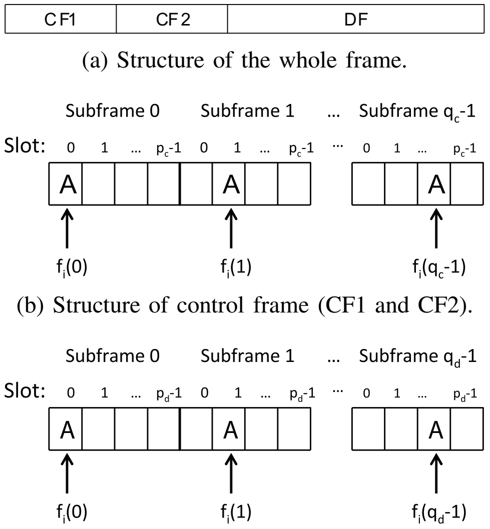
We focus on unicast communication. Each node u randomly selects one of its neighbors v as the destination of its packets in each frame. Given an arbitrary node u and the destination v of its packets, we define its contention set $C(u)$ as the set of nodes, whose transmissions and the transmission from Node u to v interfere each other. We define $C_1(u)$ as the set of Node v and the one-hop neighbors of Node v, excluding Node u itself, and $C_2(u)$ as the set of nodes, whose packets are destined either to one of the one-hop neighbors of Node u or to Node u itself. Thus, $C(u) = C_1(u) \cup C_2(u)$ and $C(u) \subset N_2(u)$. We assume that the transmission channel is error-free and a reception failure is only due to packet collisions. We assume

Manuscript received August 18, 2013. The associate editor coordinating the review of this letter and approving it for publication was L. Le.

Y. Liu and L. Zhang are with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China (e-mail: liu-ym05@mails.tsinghua.edu.cn, linzhang@tsinghua.edu.cn).

V. O. K. Li and K.-C. Leung are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: vli@eee.hku.hk, kcleung@ieee.org).

Digital Object Identifier 10.1109/WCL.2013.091113.130592



(a) Structure of the whole frame.

(b) Structure of control frame (CF1 and CF2).

(c) Structure of data frame (DF).

Fig. 1: The frame structure.

that an acknowledgment can be sent back to the transmitter once the packet is received successfully by the receiver.

III. Proposed Algorithm

Each node v is assigned a unique polynomial over Galois Field $GF(p)$ with degree k mod p, $f_v(x) = \sum_{i=0}^{k} a_i x^i (\mathrm{mod}\, p)$, where $v \in V$, as its time slot assignment function (TSAF). In each data frame and control frame, Node v transmits in the time slot $f_v(i)$ in Subframe i, where $i = 0, 1, \ldots, q-1$. $(f_v(0), f_v(1), \ldots, f_v(q-1))$ is known as time slot location vector (TSLV) for Node v [5].

For each node, CF1 is used to distribute its information, and to collect the information of its neighbors. CF2 is used to distribute the information of its one-hop neighbors collected in CF1 and to collect the transmitted information, which is received by its neighbors in CF1. The information of each node includes its own identity number (ID), TSAF, and ID of the destination of its packets. Note that the structure of control frames guarantees that there is at least one collision-free control slot in each control frame. Thus, each node can collect the information of all its two-hop neighbors after CF1 and CF2, and transmits the data packets according to the collected information, as specified in the next section. Some overhead is introduced in the proposed algorithm by implementing acknowledgments and control frames (CF1 and CF2) to distribute and collect the information of the two-hop neighbors of each node. Each node transmits its ID, the ID of the destination of its packets, and TSAF in CF1. Note that each TSAF is a degree-k polynomial, and each of its coefficients can be represented in one byte, where each coefficient is no larger than 255. Hence, each TSAF can be represented in $(k+1)$ bytes. The source and destination IDs can be represented in two bytes each. In CF2, each node broadcasts all the information received from its one-hop neighbors (up to $D_{\max}$). Hence, we need at most $(k+5)D_{\max}$ bytes to broadcast all such information. Each slot in CF1 is designed to accommodate the transmission of one packet of $(k+5)$ bytes. Each slot in CF2 is designed to accommodate the transmission of one packet of $(k+5)D_{\max}$ bytes. To simplify the calculation, each acknowledgment packet at the end of each data slot is assumed to be one byte. The overhead, defined as the portion of data transmitted in a frame that is used for control information, is as follows:

$$\begin{aligned} \beta &= \frac{(k+5)(D_{\max}+1)pq}{(L+1)pq+(k+5)(D_{\max}+1)pq}, \\ &= \frac{(k+5)(D_{\max}+1)}{L+1+(k+5)(D_{\max}+1)}. \end{aligned} \qquad (1)$$

where L is the length of the payload packet in bytes. As shown in (1), the overhead introduced in our algorithm is independent of the total number of nodes in the network, N. Thus, the overhead of our proposed algorithm does not increase with the network size.

We propose to utilize the unassigned slots in a collision-free manner, thus allowing nodes to utilize both the assigned and unassigned slots, thereby improving the average throughput.

Let $PU(u)$ be the set of conflict-free time slots that can be used for transmission from Node u (where $u = 1, 2, \ldots, N$) in the current frame. Time slot j in Subframe i is in $PU(u)$ if and only if $j \notin \bigcup_{m \in \{u\} \cup C(u)} \{f_m(i)\}$. That is, $PU(u) = \{(i,j) | 0 \leq i \leq q-1, 0 \leq j \leq p-1, j \notin \bigcup_{m \in \{u\} \cup C(u)} \{f_m(i)\}\}$.

Let M be a permutation of the vector $(1, 2, \ldots, N)$, which is known by all N nodes a priori. The priority of each node u, where $u = 1, 2, \ldots, N$, in the t-th frame, where $t = 0, 1, \ldots$, can be computed universally as follows:

$$Priority(u, t) = \mathrm{mod}((M(u) + t), N). \qquad (2)$$

Thus, each node collects the IDs and calculates the priorities of the nodes in its contention set after CF1 and CF2. The nodes with the largest priorities within their contention sets are allowed to utilize the time slots in their PUs. Note that if Node u is allowed to utilize the time slots in $PU(u)$, the transmissions from Node u to the destination in these slots are collision-free, since there are no nodes in $C(u)$ transmitting during $PU(u)$. It is obvious that the priority of each node is unique in every frame and uniformly distributed as time evolves. Thus, each node utilizes the unassigned time slots in a uniform way. The work flow of the proposed topology-transparent scheduling algorithm is described in the following:

1) Each node u calculates its TSLV based on its assigned TSAF, $f_u(x)$.

2) In CF1, each node u broadcasts its ID, TSAF $f_u(x)$, and ID of the destination of its packets according to its TSLV and stores the information received from its one-hop neighbors. In CF2, each node u broadcasts the information of its $D(u)$ neighbors received in CF1 as a packet according to its TSLV, and stores the information received from its neighbors. Hence, each node u knows the ID and TSAF of Node v, and the destination ID of the packets from Node v, where $v \in C(u)$.

3) Each node u calculates its $PU(u)$ according to the aforementioned discussion.

4) In each data frame, each node transmits packets at its assigned slots. Each node with the largest priority among its contention set also transmits its packet at the unassigned slots in its PU.

IV. Performance Analysis

The average throughput of the assigned slots is extensively studied in [5]. We study the average throughput of the unassigned slots in the following.

Consider an arbitrary node u with n nodes in its contention set. Node u cannot utilize the slots, which are the assigned slots of some nodes in its contention set and Node u itself. There are p^2 TSLVs in total. Consider an arbitrary subframe j, where $0 \leq j \leq q-1$. Let M^l denote the number of ways to select $n+1$ out of p^2 TSLVs such that the assigned slots of these $n+1$ nodes are exactly l specific slots in Subframe j, where $l = 1, 2, \ldots, p$. Note that there are $\binom{p^2}{n+1}$ ways to select $n+1$ out of p^2 TSLVs, and there are $\binom{p}{l}$ ways to select l out of p slots in one subframe. Thus, the probability that there are $p-l$ slots that can be utilized by Node u, $P(l)$, is as follows:

$$P(l) = \frac{M^l \binom{p}{l}}{\binom{p^2}{n+1}}. \tag{3}$$

As discussed in Section III, the priority of each node changes in a uniform manner as time evolves. Thus, a node with n contention nodes has the largest priority and can utilize the unassigned slots once every $n+1$ frames. We obtain the average throughput of unassigned slots, conditioned on n, $E[G_{unassigned}|n]$, as follows:

$$E[G_{unassigned}|n] = \frac{\sum_{l=1}^{p} (p-l)P(l)}{(n+1)p}. \tag{4}$$

Thus, we have:

$$E[G_{unassigned}] = \sum_{m=1}^{N-1} E[G_{unassigned}|n=m] \Pr(n=m). \tag{5}$$

However, we can hardly obtain the distribution of n. We approximate the average throughput of unassigned slots as (6) (which is found to be quite accurate when compared with the simulation results as exhibited in Section V):

$$E[\bar{G}_{unassigned}] \approx E[G_{unassigned}|\bar{n}], \tag{6}$$

where $\bar{n}$ is the average number of nodes in the contention set of a node.

We propose to estimate $\bar{n}$ as:

$$\begin{aligned} \bar{n} &\approx E[D] + (E[D]-1)E[\frac{1}{D}] + (E[D]-1)^2 E[\frac{1}{D}], && (7) \\ &\approx E[D] + \frac{E[D]-1}{E[D]} + \frac{(E[D]-1)^2}{E[D]}. && (8) \end{aligned}$$

D is the number of neighbors of a node. (8) holds, since $\frac{1}{E[D]}$ is the first order approximation of $E[\frac{1}{D}]$ [6]. Consider an arbitrary transmission from u to v. $E[D]$ represents the number of nodes including v and the neighbors of v other than u. Note that each node randomly selects one of its neighbors as its destination. Thus, $\frac{E[D]-1}{E[D]}$ represents the average number of neighbors of u other than v, the destinations of which are u. The average number of the neighbors of w (an arbitrary neighbor of u other than v), the destinations of which are w, is $\frac{E[D]-1}{E[D]}$. Node u has $E[D]-1$ such neighbors as w on the average (v is not considered here, since v and its neighbours has been counted in the first term $E[D]$). Thus, $\frac{(E[D]-1)^2}{E[D]}$ represents the average number of the other two-hop neighbors of u, the destinations of which are the neighbors of u.

The calculation of M^l, where $l = 1, 2, \cdots, p$, is as follows. Given an arbitrary subframe j, we categorize p^2 TSAFs into p different subsets F_h ($h = 0, 1, \ldots, p-1$) according to their function values in Subframe j. The function values of TSAFs in F_h ($h = 0, 1, \ldots, p-1$) are h. Note that a TSAF over $GF(p)$ is uniformly distributed over $\{0, 1, 2, \ldots, p-1\}$. Thus, $|F_h| = p$, where $h = 0, 1, \ldots, p-1$.

Let A_{p_i} (where $i = 1, 2, \ \ldots, l$) be the set of events in which none of the chosen $n+1$ TSAFs has the function value p_i in Subframe j, where $0 \leq p_i \leq p-1$. Note that the total number of TSAFs which have the function value p_i (where $i = 1, 2, \ \ldots, l$) is lp and we choose $n+1$ TSAFs from those TSAFs, the function values of which in Subframe j are not any p_i (where $i = 1, 2, \ \cdots, l$). Thus, the cardinality of the intersection of any m sets from A_{p_i} is $\binom{(l-m)p}{n+1}$, where $i = 1, 2, \ \ldots, l$ and $m = 1, 2, \ldots, l$. M^l is equal to the cardinality of the complementary set of $\bigcup_{i=1}^{l} A_{p_i}$. We define a function $F(x, y)$ which equals $\binom{x}{y}$ if $x \geq y$, and zero, otherwise. Thus,

$$M^l = F(lp, n+1) - \left| \bigcup_{i=1}^{l} A_{p_i} \right|, \tag{9}$$

where $l = 1, 2, \ \cdots, p$.

Applying the Inclusion-Exclusion Principle, we can obtain the following results:

$$\begin{aligned} M^l \; &= \; F(lp, n+1) \\ &\quad - \sum_{m=1}^{l} (-1)^{m-1} \binom{l}{m} F((l-m)p, n+1), && (10) \end{aligned}$$

where $l = 2, \cdots, p$ and $M^1 = F(lp, n+1)$.

V. Performance Evaluation

We study the average throughput of our algorithm, which is defined as the average number of packets transmitted successfully per node per time slot. We compare our algorithm with the PMNF coloring algorithm [7] (referred to as coloring algorithm), the topology-transparent scheduling algorithm [5] (referred to as Ju's algorithm), in which only assigned slots are utilized, and an ideal conflict graph coloring algorithm. In this ideal conflict graph coloring algorithm, a central scheduler knows the network topology information and the destination of the packets originated from each node, constructs a conflict graph, and applies the distance-1 vertex coloring algorithm to compute and distribute the transmission schedules to each node. When the destination of the packets of an arbitrary node changes, we assume that the central scheduler can get the changed information, re-compute, and distribute the updated schedules immediately without introducing any overhead. Thus, this ideal algorithm is impractical.

We conduct simulation under the geometric model for the average performance. In the geometric model, all nodes are distributed uniformly and randomly in a region A of 1000 m × 1000 m. Given $D_{\max}$, we set the interference range of each node R_I such that the probability that the number of interfering neighbors of an arbitrary node exceeding $D_{\max}$, which

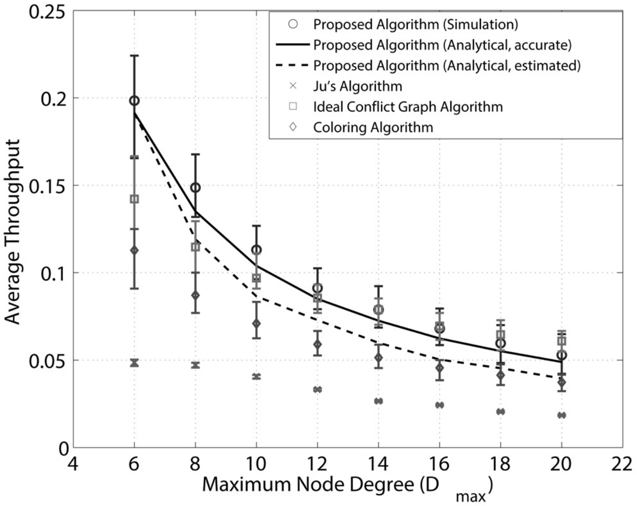


Fig. 2: Average throughput vs. maximum node degree.

is $\sum_{i=D_{\max}+1}^{N-1} \binom{N-1}{i}(\frac{\pi R_I^2}{A})^i(1-\frac{\pi R_I^2}{A})^{N-1-i}$, is smaller than 0.05.[1] For example, $R_I = 87$ m if $(N, D_{\max}) = (256, 10)$. If there exist more than $D_{\max}$ nodes in the interference range of a node, the nodes other than $D_{\max}$ randomly selected interfering nodes are assumed to be non-interfering. This guarantees that the maximum node degree is $D_{\max}$. We set L to 1024 bytes and the overhead of our algorithm can be calculated according to (1). We assume a heavy traffic condition with all users backlogged. For each result, we run each simulation for 100 randomly generated topologies. For each data point, the 95% confidence intervals are also drawn in the figures.

Given that $N = 256$, we investigate the average throughput of our algorithm with $D_{\max}$ varying from six to 20. Each analytical result shown in Fig. 2 is the sum of the analytical results of the assigned slots obtained in [5] and the unassigned slots obtained in (6). We can observe that the average throughput of our algorithm is better than that of the coloring algorithm in [7]. Moreover, the average throughput of our algorithm is almost the same as that of the ideal conflict graph coloring algorithm. Indeed, for some cases, our algorithm even performs better. This can be explained as follows. The throughput of the ideal conflict graph coloring algorithm is determined by the number of colors needed to color the neighborhood of the node with the largest node degree. Nodes with smaller node degrees have the same throughput as those nodes with the largest node degree. However, there are more unassigned slots to be utilized for a node with node degree less than $D_{\max}$ in our algorithm. Our algorithm allows those nodes to utilize more unassigned slots, but the ideal conflict graph coloring algorithm does not. The average node degree is typically smaller than the maximum node degree $D_{\max}$ in reality. Thus, we conclude that the performance of our algorithm is very close to, or sometimes even better than, that of the ideal conflict graph coloring algorithm, even in static multihop networks. Considering the overhead introduced by deploying the ideal conflict graph coloring algorithm, our algorithm obviously performs much better, since in reality the ideal conflict graph coloring algorithm has to spend a lot of time and network resources to collect information, compute, and distribute the updated schedules when the conflict graph changes [8], and is thus impractical.

The validation of our analytical results is also shown in Fig. 2. We can observe that our simulation results yield a better performance than the analytical ones. This is mainly due to the fact that the estimated value of $\bar{n}$ is larger than that in reality, since some nodes may be both one-hop and two-hop neighbors of a particular node and are counted more than once as shown in (8). When the average number of contention nodes is known, our analytical results become more accurate. As shown in Fig. 2, using the actual value of the average number of contention nodes, our analytical results match well with the simulation results. The observed discrepancy is incurred due to the approximation made for the calculation of the average throughput in (6).

VI. Conclusion

We propose to improve the throughput of topology-transparent scheduling algorithms by utilizing the unassigned slots in a collision-free manner, based on the schedules of the assigned slots for contending users. The performance of our algorithm is much better than the existing topology-transparent scheduling algorithms. More importantly, the average throughput of our algorithm is better than that of some topology-dependent (coloring) algorithms even in static multihop networks. Hence, the proposed topology-transparent scheduling algorithm can be a good choice in both static and mobile multihop networks.

Acknowledgment

This research is supported in part by the Research Grants Council of the Hong Kong, under Grant No. HKU 714310E.

References

[1] Z. Cai, M. Lu, and C. Georghiades, "Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 970–984, Jul. 2003.

[2] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, pp. 23–29, Feb. 1994.

[3] A. M. Chou and V. O. K. Li, "Fair spatial TDMA channel access protocols for multihop radio networks," in *Proc. 1991 IEEE INFOCOM*, pp. 1064–1073.

[4] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *Proc. 2009 IEEE INFOCOM*, pp.226–234.

[5] J. H. Ju and V. O. K. Li, "An optimal topology-transparent scheduling method in multhop packet radio networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 3, pp. 298–306, Jun. 1998.

[6] A. Papoulis and S. U. Pillai, Probability, *Random Variables, and Stochastic Processing*, pp. 112–113. McGraw-Hill Education. 2002.

[7] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, Mar. 1999.

[8] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: distributed randomized TDMA scheduling for wireless ad hoc networks," *IEEE Trans. Mobile Computing*, vol 8, no. 10, pp. 1384–1396, Oct. 2009.

[9] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 130–137, Jun. 2001.

[1] $E[D]$ can be calculated as $E[D] = \frac{N\pi R_I^2}{A}$.