Increasing the Throughput of Multihop Packet Radio Networks with Power Adjustment *

Chi-Fu Huang, Yu-Chee Tseng, Shih-Lin Wu, and Jang-Ping Sheu

Abstract-

The Packet Radio Network (PRN) is an attractive architecture to support mobile and wireless communication. Although the code assignment problem has been studied extensively on PRN, we observe in this paper that the *power control* problem has been ignored by most works, but may have significant impact on performance. By *power control*, we mean that the transmission ranges of stations are tunable. We show given a PRN in which each host already received a code, how to adjust the powers of stations to control/improve the topology of the PRN without violating the original code assignment. Several schemes are proposed. Through simulations, we demonstrate that although the code assignment problem is NP-complete and thus computationally very expensive, using our power adjustment schemes can easily improve the network performance by about 20% with polynomial costs.

I. INTRODUCTION

Packet Radio Networks (PRN) were first demonstrated in 1969 at the University of Hawaii [2] and since then have greatly increased their presence and importance for computer communications. A PRN consists of a number of *stations* placed in a geographically distributed area, where each station has a computer and a transceiver. Two stations are said to be *connected*, if their radio transceivers can communication with each other directly. A PRN can be considered as a graph with a certain topology, and to reflect the fact that two stations may have to communicate indirectly by relaying stations, we will sometimes refer to it as a *multi-hop* PRN. PRNs have applications in areas where wireline networks are difficult to deploy.

One widely studied issue for PRN is the *code assignment problem* [3, 5, 6, 11, 12, 14, 17], where each host should be assigned a collision-free code for transmission. A tree-based scheme is proposed in [5], where it is also shown that determining the least number codes for any network is NP-complete. The authors also proposed a distributed version, which uses a traveling token. Another scheme which also uses traveling token is [6], where the common channel is split into a control segment and a transmission segment. The control segment is to avoid conflicts among hosts and to increase the utilization of the transmission segment. Heuristics are proposed in [3, 12, 14] for code assignment in regular and general PRN topologies. A transmitter-oriented code assignment is presented in [12]. Reference [3] assigns codes based on hosts' degrees. [14] chooses an unassigned host to be processed first if it has most neighbors already receiving codes. The concept of *maximum independent set* is used in [17] for broadcast scheduling; the result can also be used for code assignment.

The above results are suitable for traditional PRNs with low or no mobility. Some recent protocols start to tolerate mobility [8, 7, 16]. The protocol in [8] employs a polling mechanism. Once polled, an intending sender will use its sending code to transmit. In [7], the protocol assigns channels to hosts statically. It mandates that the channel assigned to a host must be different from those of its two-hop neighbors. A *hop-reservation* MAC protocol based on very-slow frequency-hopping spread spectrum is proposed in [16].

Although the code assignment problem has been studied extensively on PRN, we observe in this paper that the *power control* problem has been ignored by most works, but may have significant impact on performance of PRN. By *power*, we mean the transmission strength (or range) of stations. Let's raise two extreme cases. If the transmission power is tuned to the minimum possible level, we may encounter very weakly connected networks with long routing paths between stations and many traffic bottleneck stations (such as at *articulation points*). On the contrary, tuning powers too high will result in too strongly connected networks, where radio signals can easily contend and collide with each other, leading to little spatial reuse. Further, it is not necessary that all stations use the same power level; the possibility of tuning individual stations' powers also deserves study.

The purpose of this paper is not to propose a new code assignment solution. Instead, we show that given a PRN in which each host already received a code, how to adjust the powers of stations to obtain a "better" network without violating the original code assignment. In some sense, we try to control/improve the topology of the PRN by tuning powers. So our result can be regarded as building on top of those code assignment solutions. A *distance-based* scheme and a *degree-based* scheme are proposed. On top of these, we also introduce a *code randomization* mechanism to further improve performance. Through simulations, we demonstrate that although the code assignment problem is NP-complete and thus computationally very expensive, using our power adjustment schemes can easily improve the network performance by about 20% with polynomial costs.

Some works have addressed the power issue, but on different environment. Assuming a contention-based channel model (such as ALOHA or CSMA), [15] and [10] show how to determine the optimmal transmission ranges in PRN, where hosts'

---

[*] This work is supported by the National Science Council of the Republic of China under Grant # NSC89-2213-E-008-024 and #NSC89-2213-E-008-025, and the Ministry of Education, the Republic of China, under grant 89-H-FA07-1-4 (Learning Technology). C.-F. Huang and Y.-C. Tseng are with the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan; S.-L. Wu is with Department of Electrical Engineering, Chang Gung University, Taiwan; J.-P. Sheu is with the Department of Computer Science and Information Engineering, National Central University, Taiwan. (contact email: yctseng@csie.nctu.edu.tw)

0-7803-7128-3/01/$10.00 (C) 2001 IEEE

powers can be equal and non-equal, respectively. Reference [13] also considers topology control; its goal is to obtain a connected or bi-connected network such that the maximal power used by all hosts is minimum. Energy-efficient communication for sensor networks is addressed in [9]. In the area of mobile ad hoc networks (MANET), power issues have been studied on the MAC layer [18] and routing layer [4].

The remainder of the paper is structured as follows. Section II. gives the preliminaries and motivations, followed by formal definitions of the power adjustment problems to be solved in this paper. Section III. proposes several centralized power-adjustment schemes assuming that only a single code is assigned to each station. Section IV. extends to the assumption that multiple codes can be assigned to a host. Simulation results are in Section V.. Finally, conclusions are drawn in Section VI..

II. PRELIMINARIES AND MOTIVATIONS

A PRN consists of a number of stations. Based on the connectivity between stations, it will form a certain network topology. Since this work focuses on power adjustment, the connectivity may not be a fixed parameter, and will depend on the transmission powers of stations. Following the *sender-based* assumption in many works [3, 11, 14, 17], we assume that each station accesses, and thus transmits, based on the code assigned to it. A code is a resource that a host can use freely without interference; it is a logical term and could be a *time slot* in TDMA, a *frequency band* in FDMA, and an *orthogonal code* in CDMA. When assigning codes to stations, we need to consider two types of collisions: *primary* and *secondary*. A primary collision occurs when two stations using the same code can hear each other, while a secondary collision occurs when two stations using the same code can be heard by a third common station (or sometimes known as the *hidden-terminal problem*).

The code assignment problem is to obtain codes to stations while avoiding the primary and secondary collisions such that the total number of codes used is minimal. This problem has been shown to be NP-complete[3].

However, the above referenced works all assumed that the PRN topology is a given fixed input, based on which code assignment has to be solved. Since transmission powers are adjustable in this paper, we will assume that we are given a PRN with only stations' locations. Based on the yet-to-be-determined powers of stations, the network topology will change accordingly. Also, it is natural and practical to assume that a transmission power level should not be infinitely small or large, but should fall in a range $[P_{min}, P_{max}]$.

Below, we give the formal problem statements. Note that in the definitions both code assignment and power control will be involved.

Definition 1 **Single-Code Assignment with Power Adjustment (SAPA):** Given a set of mobile stations $\{H_1, H_2, \ldots, H_n\}$, where each station H_i is placed in a location $L_i, i = 1..n$, our goal is to assign each host $H_i, i = 1..n$, a code based on the sender-based rule and a transmission power level P_i satisfying $P_{min} \leq P_i \leq P_{max}$ such that the network throughput is maximized.

Definition 2 **Multi-Code Assignment with Power Adjustment (MAPA):** Given a set of mobile hosts $\{H_1, H_2, \ldots, H_n\}$, where each station H_i is placed in a location $L_i, i = 1..n$, and a set of integers $\{t_1, t_2, \ldots, t_n\}$, our goal is to assign each host H_i a set of t_i codes based on the sender-based rule and a transmission power level P_i satisfying $P_{min} \leq P_i \leq P_{max}$ such that the network throughput is maximized.

Note that MAPA is an extension of SAPA by allowing more than one codes for each station. The motivation is to take into consideration the difference of traffic loads among stations. When all c_i's are equal, MAPA degenerates to SAPA. Given any network, the code assignment problem has been proved to be NP-complete even when all hosts share a common transmission power [3]. If we impose that $P_{min} = P_{max}$, SAPA will be reduced to the code assignment problem. Thus, SAPA and MAPA, which extend the code assignment problem, are both computationally intractable.

III. SOLUTIONS FOR SAPA

In this section, we propose a centralized solution. We assume that the powers to be used by stations are computed by a central station, which knows the locations of all stations. The steps are outlined below.

1. Pick an *initial power* T, where $P_{min} \leq T \leq P_{max}$.
2. For $i = 1..n$, let $P_i = T$. Based on this power setting, construct a graph corresponding to the topology of the PRN.
3. Apply any heuristic for code assignment on the current topology of the PRN.
4. Perform our power adjustment scheme based on G (see the subsequent sections).

In the above steps, we first choose a common initial power T for each station, where T is an input parameter. By this setting, a code assignment scheme is involved to determine a code for each station. Then we perform power adjustment to increase hosts' powers, if possible. However, note that the power adjustment in step 4 is limited by the constraint that the already-determined code assignment in step 3 will not cause any primary and secondary collision. Also note that how to choose the best value of T may not be an easy job. In our approach, we will use simulation to determine the best T by hopping through the interval $[P_{min}..P_{max}]$, and we will use network throughput as our metric for comparison.

In the following, we propose several ways for step 4.

A. *Distance-Based Scheme*

From the above steps, we already obtain a network G, in which each host i has a common power P_i and a code c_i. In the distance-based scheme, we will greedily increase the powers of individual stations to increase the network connectivity. By *network connectivity*, we simply count the number of links in the graph. The intuition is that a network with more links may have higher throughput. However, doing so is under the constraint that no primary and secondary collision should occur.

The scheme works as follows. We first collect all station pairs that are not connected in G. These pairs are sorted in an ascending order according to their distances. Then we sequentially check each pair in the list and try to add it into G (by increasing the two corresponding stations' transmission powers). This is repeated until no more pair can be added. In the following steps, the distance of hosts i and j is represented by $dist(i, j)$, and the minimum transmission power required for two hosts distanced by d to communicate is denoted by $\lambda(d)$.

a) Let L be the list of all station pairs (i, j) such that link $(i, j) \notin G$ and $\lambda(dist(i, j)) \leq P_{max}$. Sorted L in an ascending order of the distance between the two hosts in each pair.

b) Define a collision array $col[1..n]$ such that $col[i]$ is the set of codes used by host i itself and all neighbors of i in G, i.e.,

$$col[i] = \{c_i\} \cup \{c_j | (i, j) \in G\}.$$

Intuitively, any host who is not adjacent to i in G and who intends to establish a link with i must not use any code in $col[i]$; otherwise, primary or secondary collision will occur.

c) Pick the first entry (i, j) in L. Otherwise, check the following two conditions:

$$\forall k : \lambda^{-1}(P_i) < dist(i, k) \leq dist(i, j) \Longrightarrow c_i \notin col[k]$$
$$\forall k : \lambda^{-1}(P_j) < dist(j, k) \leq dist(i, j) \Longrightarrow c_j \notin col[k].$$

If both conditions hold, this means that adding a link between hosts i and j will not suffer from primary and secondary collisions. If so, we change the power setting by letting $P_i = P_j = \lambda(dist(i, j))$ and update the collision array as follows:

$$col[k] = col[k] \cup \{c_i\}$$
$$\quad \text{for all } k \text{ such that } \lambda^{-1}(P_i) < dist(i, k) \leq dist(i, j)$$
$$col[k] = col[k] \cup \{c_j\}$$
$$\quad \text{for all } k \text{ such that } \lambda^{-1}(P_j) < dist(j, k) \leq dist(i, j)$$

Intuitively, the first two equations update the collision array of i and j, while the last two do for those interfered by i's and j's higher powers.

d) Remove (i, j) from L. If L is not empty, go to step c.

For example, Fig. 1(a) shows a PRN, where the circles (all of the same radius) indicate the transmission distances of hosts. The current network topology G is shown in Fig. 1(b), where the number associated with each host is the code assigned to it. After calculating list L, the following trials will be made. Note that there are some subtleties in trials 3 and 5 deserving attention.

1. Failure on link (H, E): Increasing host H's power can be granted, but increasing host E's power will cause a secondary collision at host H.

2. Failure on (F, C): Increasing F's power will cause a secondary collision at C.

3. Failure on (E, B): Increasing B's power can be granted. But increasing E's power will cause a secondary collision at H (this is because $dist(E, H) < dist(E, B)$).

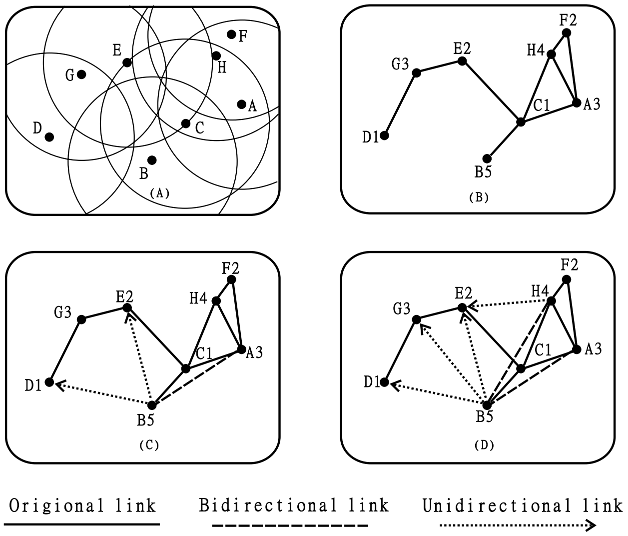

Figure 1: An example of the distance-based scheme: (a) the original G and hosts' transmission ranges, (b) the induced G and code assignment, (c) topology after adding link (B, A), and (d) topology after adding link (H, B).

4. Fail on (D, B): Increasing D's power causes a secondary collision at B.

5. Success on (B, A): Both collision tests will pass. So a link will be added between B and A, as shown in Fig. 1(c). Array entries $col[B]$ and $col[A]$ should be updated properly. However, increasing B's and A's powers will cause two directional links be added from B to D and B to E, as shown in the dotted arrows in Fig. 1(c). So the code 5 used by host B should be added to $col[D]$ and $col[E]$ too.

6. Failure on (F, E): Primary collisions occur at both ends.

7. Failure on (E, D): Secondary collision occurs at E.

8. Failure on (G, B): Secondary collision occurs at B.

9. Failure on (G, C): Secondary collisions occur at both ends.

10. Failure on (E, A): Secondary collisions occur at both ends.

11. Success on (H, B): Both collision tests pass. So tune H's and B's powers and add a new link between H and B, as shown in Fig. 1(d). Four array entries $col[B]$, $col[E]$, $col[G]$, and $col[H]$ should be updated.

12. The rest of trials will all fail.

The resulting network is shown in Fig. 1(d). We have added two bi-directional links and four uni-directional links to the network. As can be seen, host C, which was originally an articulation point and could be a heavily loaded bottleneck, is now not so any more. This is expected to relieve the network congestion significantly. We call the uni-directional links **side effect edges** and will not use them in this paper. However, how to use them is application-dependent (e.g., some routing protocols can handle such problem).

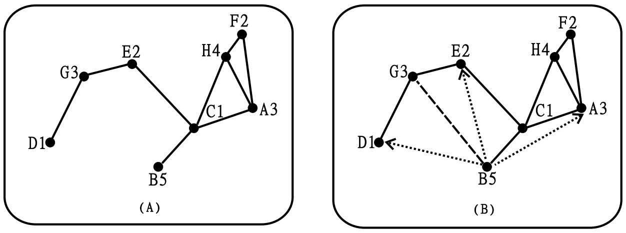

Figure 2: Examples of degree base scheme : (a)Original topology (b)Topology performed by degree base scheme

B. Degree-Based Scheme

The previous scheme uses distance as the metric to determine which link should be checked and added into the network first. In this section, we propose to use hosts' degrees in G as the metric. The rationale is that hosts with lower degrees are weaker in communication capability and thus may become bottlenecks in the network. Thus, adding links for them is more important than for those with higher connectivity.

Note that similar to the distance-based scheme, the degree-based scheme can only increase powers of hosts when no collision will occur. So the total number of codes will not be changed.

The process to adjust powers is similar to the previous scheme, except that the order that the potential links are checked is different. So we only briefly summarize the steps as follows.

a) L is still the set of potential links to be added, but is sorted differently by using host degree as the primary key, and distance as the secondary key, both in an ascending order. Note that since each pair $(i, j) \in L$ has two hosts, the lower value of the degrees of i and j are used for sorting.

b) Calculate the collision array (same as the distance-based scheme).

c) Pick the first $(i, j) \in L$ for possible power adjustment (same as the distance-based scheme).

d) Remove (i, j) from L. Also, if link (i, j) is added into the network in step c, we should properly adjust the positions of all remaining links' in L which are incident to i or j (since both i's and j's degrees have been increased by one). Then go to step c, if necessary.

Fig. 2 shows an example based on the same network in Fig. 1. Based on degrees, the following sequence of trials will be made:

1. Failure on (B, E), (D, B), (E, H), and (F, C).
2. Success on (G, B).
3. Failure on (A, B), (G, C), (H, E), and (C, F).

In this example, only one bi-directional link and three uni-directional links are added. Although the number of links being added to the network is less than what we have done earlier by the distance-based scheme, this shows a different flavor of the degree-based scheme — it tries to make weaker hosts stronger, in terms of their connectivity. In fact, in our simulations (to be shown later), we do find many situations where this scheme will outperform the distance-based scheme.

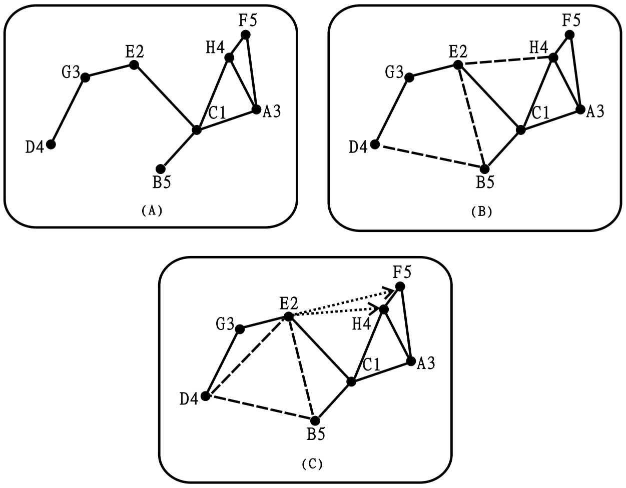

Figure 3: Applying code randomization before power adjustment: (a) a new assignment by changing hosts D's and F's codes, (b) the topology after performing the distance-based scheme, and (c) the topology after performing the degree-based scheme.

C. Code Randomization for the Distance- and Degree-Based Schemes

In this subsection, we propose a simple mechanism that can be added on top of the above two schemes to improve their performance. The main idea is to change the codes assigned to hosts in step 3. Let's use an example to motivate the idea. Observe Fig. 3, which represents the same network G in Fig. 1 and Fig. 2, but with different code assignment. Specifically, the codes used by hosts D and F are changed to 4 and 5, respectively. If we use this network as G and apply the distance- and degree-based schemes on it, three more links will be added to the network, as shown in Fig. 3(b) and Fig. 3(c), respectively. This improves the connectivity of the network as opposed to the earlier examples, which shows the potential benefit of changing codes.

Indeed, as we surveyed several code assignment algorithms in the literature [3, 14], there is a tendency of favoring using some set of codes than the others. The reason is quite obvious — the goal of code assignment is to use as few codes as possible. Thus, the same code is likely to be used by hosts that are physically close to each other, so that the code usage pattern can be as compact as possible. However, this is disadvantageous to our power adjustment, because when adding links, there will be more chance to find primary or secondary collisions.

Recall that after step 3 we already have a network G in which each host has a code $c_i, i = 1..n$. Here we propose a simple *randomization technique* to change the code assignment. We sequentially pick each host in G and try to re-select for it a new code that is not used by any of its two-hop neighbors. This will disturb the compact code usage pattern in the original assignment. However, the total number of codes used is not increased. The procedure is formally presented below. Note that this procedure can be applied on top of the distance- and degree-based schemes, and should be run before these schemes are run.

i) Let C be the set of all codes used by the network after step 3.

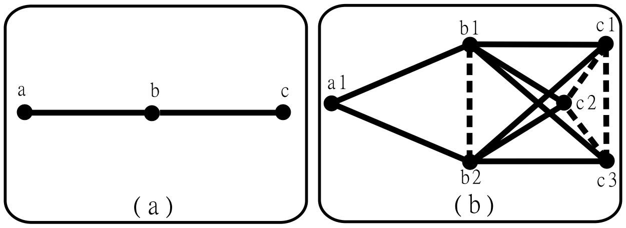

Figure 4: An example of reduction from a MAPA problem to a SAPA problem.

ii) Sequentially pick each host in an arbitrary order. For each host i, randomly pick a code from the set

$$C - \{c_j | \text{host } j \text{ is a 1-hop or 2-hop neighbor of } i \text{ in } G\}.$$

Then change c_i to this randomly selected code.

IV. EXTENDING TO MAPA WITH POWER ADJUSTMENT

In MAPA, each host can request more than one code. This is to take the difference of traffic loads among hosts into consideration. In this section, we show how to extend our result from SAPA to MAPA.

The main technique is a reduction from a graph representing a SAPA problem to one representing a MAPA problem. Let's represent a PRN of a certain topology as an undirected graph $G_m = (V_m, E_m)$, where $V_m = H_1, H_2, \ldots, H_n$ is the host set and E_m is the link set (subscript m represents "multi-code"). In V_m, each host H_i requires t_i codes. Now we translate this problem to a graph $G_s = (V_s, E_s)$ representing a SAPA problem (subscript s represents "single-code"). Specifically, for each host $H_i \in V_m$, we introduce the following t_i hosts into V_s:

$$H_{i,1}, H_{i,2}, \ldots, H_{i,t_i}.$$

Also, the link set is

$$\begin{aligned} E_s = \\ & \{(H_{i,k_1}, H_{i,k_2}) | H_i \in V_m, 1 \le k_1 \le t_i, \\ & \qquad 1 \le k_2 \le t_i, k_1 \ne k_2\} \\ & \cup \{(H_{i,k_1}, H_{j,k_2}) | (H_i, H_j) \in E_m, H_i \in V_m, H_j \in V_m, \\ & \qquad 1 \le k_1 \le t_i, 1 \le k_2 \le t_j\} \end{aligned}$$

Intuitively, in G_s, each host $H_{i,k}, k = 1..t_i$, requires one code. These t_i hosts, which represent H_i in G_m, will together require t_i codes. In the definition of E_s, the first set establishes a clique among hosts $H_{i,k}, k = 1..t_i$, which means that the codes assigned to these t_i hosts should be distinct. The second set establishes a link between each pair of H_{i,k_1} induced by $H_i \in V_m$ and H_{j,k_2} induced by $H_j \in V_m$, which indicates the fact that the two vertices H_{i,k_1} and H_{j,k_2} are physically adjacent.

Fig. 4(a) shows an example, where we are given a network of three hosts a, b, and c each requiring 1, 2, and 3 codes, respectively. Then from these three hosts we will introduce three sets of hosts $\{a_1\}$, $\{b_1, b_2\}$, $\{c_1, c_2, c_3\}$, respectively. Each set forms a clique. Also, from any host in one set, there is a link to any host in another set (so there are 6 links between $\{b_1, b_2\}$ and $\{c_1, c_2, c_3\}$). The resulting graph is shown in Fig. 4(b).

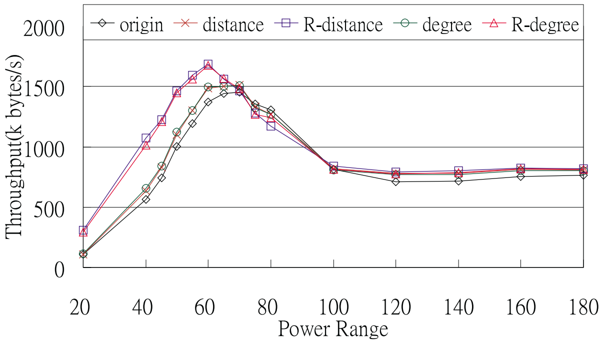

Figure 5: Effect of power adjustment at different initial power settings ($n = 50$ and $r = 1/50$).

Theorem 1 *If a code assignment to the graph G_s is minimum, the code assignment to its corresponding graph G_m is also minimum.*

With the above reduction, power adjustment on MAPA can proceed similar to that in SAPA. Given a MAPA problem, we can first tune an equal power for all hosts (by hopping from P_{min} to P_{max}, as in Section III.). For each (equal) power level, we can reduce the MAPA problem to a SAPA problem to obtain a code assignment. Then we can proceed with the power adjust schemes as we have discussed in Section III. (note that in this step, we can go back to the MAPA domain to solve the problem, except that now when checking primary and secondary collisions, multiple codes may need to be checked for a single host).

V. SIMULATION RESULTS

We have developed a simulator to evaluate the effectiveness of power adjustment. A network of n stations randomly spread in a 500×500 area was simulated. The transmission distance of each host is tunable, but no larger than 200 units. In the simulations, we adopted the heuristic *SATURATION-DEGREE-CODE-ASSIGNMENT* in [14] as our code assignment algorithm. In this scheme, hosts are given codes based on some priority. A host finding more codes being occupied by its neighbors has the highest priority. Ties are broken by preferring hosts with more neighbors already owning codes.

A TDMA channel model was used, where each time slot is 20 μs. The transmission rate is 10 Mbps, so 200 bits can be sent in one slot. Data packets, each of size 2K bytes, were generated with an arrival rate of r to the network. Each packet had a randomly chosen source and destination. For each packet, the Dijkstra's shortest-path algorithm was used to choose routes. Note that although unidirectional links exist, we only use bidirectional links for transmission for reasons of being practical. We measured the network throughput, in an end-to-end semantic. Only packets that successfully reached their destinations were counted. All results were from average of 100 random networks each being run for 100 seconds of simulation time.

Fig. 5 is to demonstrate step-by-step how our schemes work. First we pick a common transmission distance for all hosts. This value is to reflect the input parameter T in our scheme. Then we try to adjust hosts' powers using our schemes. The "Power

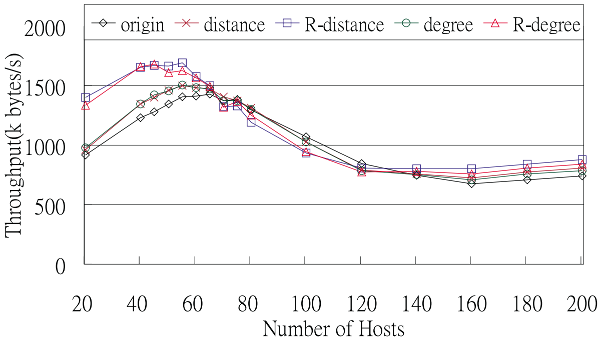

Figure 6: Network throughput at different host densities (transmission range = 60∼200 and $r = 1/50$).

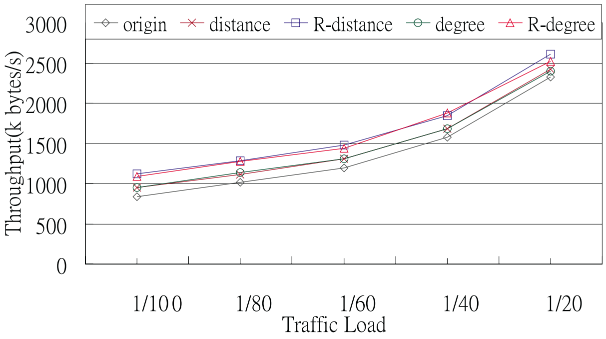

Figure 7: Throughput vs. traffic Load (transmission range = 60∼200 and $n = 50$)

Range" in the x-axis is the input common transmission distance for all hosts. The "original" curve is the network throughput without power adjustment. The other four curves are from our schemes. In the figure, an initial "R" means that the code randomization mechanism is adopted. So the power settings correspond to the points with peak performance are what will be discovered by our schemes.

It is worth looking at the network throughput at different initial transmission distances. Power adjustment is more effective at smaller distances, which represents sparser networks. As the distance increases up to a certain level (e.g., 70), the benefit disappears. This is because the network becomes denser (with many links). So adding more links is less beneficial.

Fig. 6 shows the effectiveness of power adjustment at different host density (a larger n means higher density). Each number represents the best power setting for the corresponding scheme. Generally, when the network is less dense, schemes with code randomization performs the best, which are followed those without code randomization, which are followed by that without power adjustment. Similar to the earlier observation, when the network is dense up to a certain level (around 80 hosts in a 500×500 area), power adjustment will not help.

In Fig. 7, we further vary the traffic load. As can be seen, power adjustment improves performance in all range of loads.

VI. CONCLUSIONS

Power control is an important issue in almost all kinds of wireless architectures. This paper has developed several schemes to improve the topology of a PRN through power control. We have successfully applied our results on top of earlier code assignment solutions. Interestingly, we demonstrate that although code assignment is a computationally very expensive job, it does not prohibit us from improving the performance of a PRN through power control with polynomial costs. In addition, we also show how to reduce a multi-code assignment problem to a single-code assignment problem and then use the proposed power adjustment schemes to improve the network performance.

REFERENCES

[1] Metricom. http://www.metricom.com, Los Gatos, CA.

[2] N. Abrahmson. The ALOHA system - Another alternative for computer communications. In *Proc. of FJCC*, pages 281–285, 1970.

[3] A. A. Bertossi and M. A. Bonuccelli. Code assignment for hidden terminal interference avoidance in multihop packet radio networks. *IEEE/ACM Trans. on Networks*, 3(4):441–449, Aug. 1995.

[4] J.-C. Cano and P. Manzoni. A Performance Comparison of Energy Consumption for Mobile Ad Hoc Network Routing Protocols. In *Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2000.

[5] I. Chlamatic and S. Kutten. Tree-Based Broadcasting in Multihop Radio Networks. *IEEE Trans. on Comput.*, Oct. 1987.

[6] I. Cidon and M. Sidi. Distributed Assignment Algorithms for Multihop Packet-Radio Networks. In *Proc. of INFOCOM '88*, 1988.

[7] J. J. Garcia-Luna-Aceves and J. Raju. Distributed Assignment of Codes for Multihop Packet-Radio Networks. In *Proc. of MILCOM '97*, Nov. 1997.

[8] Z. J. Hass. On the Performance of a Medium Access Control Scheme for the Reconfigurable Wireless Networks. In *Proc. of MILCOM '97*, 1997.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Hawaii Conf. on System Sciences*, 2000.

[10] T. C. Hou and V. O. Li. Transmission range control in multihop packet radio networks. *IEEE Trans. Commun.*, 34(1):38–44, Jan. 1986.

[11] L. Hu. Distributed code assignment for cdma packet radio networks. *IEEE/ACM Trans. on Networks*, 1(6):668–677, Dec. 1993.

[12] T. Makansi. Transmitter-Oriented Code Assignment for Multihop Radio Networks. *IEEE Trans. Commun.*, COM-35(12):1379–82, 1987.

[13] R. Ramanathan and R. R. Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. In *INFOCOM 2000*, 2000.

[14] A. A. B. Roberto Battiti and M. A. Bonuccelli. Assigning codes in wireless networks:bounds and scaling properties. *ACM/Baltzer Wireless Networks*, 5:195–209, 1999.

[15] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. Commun.*, 32(3):246–257, Mar. 1984.

[16] Z. Tang and J. J. Garcia-Luna-Aceves. Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks. In *Proc. of INFOCOM '99*, 1999.

[17] T. H. Vuong and D. T. Huynh. Broadcast scheduling in packet radio networks. In *ICCCN 98*, 1998.

[18] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control. *IEEE Journal on Selected Area in Communications*, 18(9):1647–57, 2000.