Distributed Adaptive Power Control in IEEE 802.11 Wireless Networks

Wang Hei Ho and Soung Chang Liew
Department of Information Engineering
The Chinese University of Hong Kong
{whho4, soung}@ie.cuhk.edu.hk

Abstract — *Optimizing spectral reuse is a major issue in large-scale IEEE 802.11 wireless networks. Power control is an effective means for doing so. Much previous work simply assumes that each transmitter should use the minimum transmit power needed to reach its receiver, and that this would maximize the network capacity by increasing spectral reuse. It turns out that this is not necessarily the case, primarily because of hidden nodes. In a network without power control, it is well known that hidden nodes give rise to unfair network bandwidth distributions and large bandwidth oscillations. Avoiding hidden nodes (by extending the carrier-sensing range), however, may cause the network to have lower overall network capacity. This paper shows that in a network with power control, reducing the instances of hidden nodes can not only prevent unfair bandwidth distributions, but also achieve higher overall network capacity compared with the minimum-transmit-power approach. We propose and investigate two distributed adaptive power control algorithms that minimize mutual interferences among links while avoiding hidden nodes. In general, our power control algorithms can boost the capacity of ordinary non-power-controlled 802.11 networks by more than two times while eliminating hidden nodes.*

Keywords — Wireless Networks, WLAN, Power Control, 802.11, Network Capacity, Scalability, CSMA/CA, Ad-hoc Networks, Hidden Nodes, Exposed Nodes.

I. INTRODUCTION

Optimizing spectral reuse is a major issue in large-scale IEEE 802.11 wireless networks. Power control is an effective means for doing so, and can allow 802.11 and 802.11-like wireless networks to achieve scalable capacity [1]. Much of the previous work on power control focuses on maximizing spectral reuse by minimizing the transmit powers of nodes [2] [3] [4]. However, such power control algorithms may not be desirable in that they overlook the effect of hidden nodes (HN), which may give rise to unfair network bandwidth distributions and bandwidth oscillations [5] [6].

HN can be eliminated by extending the carrier-sensing range [5]. However, doing so may cause the exposed-node (EN) problem in which links that do not otherwise interfere with each other are barred from simultaneous retransmission because they are within the carrier-sensing range of each other. In a *non-power-controlled network*, there is generally a tradeoff between HN and EN, which translates to a tradeoff between fairness and overall network capacity.

For *power-controlled networks*, the minimum-transmit-power approach aims to reduce EN. However, it cannot avoid HN, and may actually incur a higher level of HN. This paper shows that judicious power control can reduce EN while avoiding HN altogether. In particular, not only the usual bandwidth distribution and oscillation problems associated with HN can be eliminated, the overall network capacity can also be higher than that in the minimum-transmit-power approach.

This paper proposes and investigates two distributed adaptive power control algorithms in which the transmit powers of transmitters are adapted to the positions of their surrounding links in addition to the connectivity requirements with their receivers. These algorithms make sure that (i) links that do not mutually interfere with each other remain non-interfering, and that existing interfering links may be made to be non-interfering, after power adjustments; and (ii) new hidden nodes will not be created.

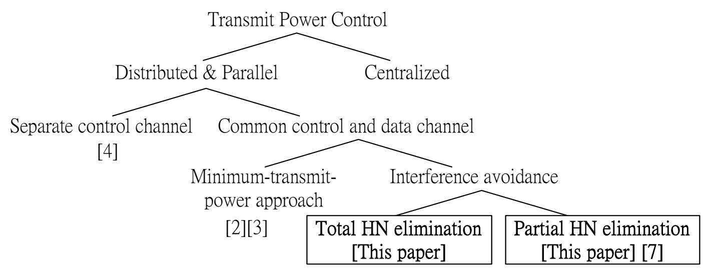


Figure 1. Classification of related work on transmit power control.

Related Work

Figure 1 shows a possible classification of various approaches for power control and provides the context under which our work was performed. Most previous investigations adopt the minimum-transmit-power approach (e.g., [2] and [3]). The COMPOW protocol in [2] selects a *common* minimum transmit power for all nodes such that network connectivity is preserved. Essentially, the transmit powers of all nodes are set to the maximum of the minimum power requirements of all links. In the CLUSTERPOW protocol [2], on the other hand, transmit powers of nodes may vary, and each transmitter forwards packets using the smallest powers required to reach their respective receivers. Reference [3] is similar in that it proposed a distributed power-control algorithm that allows nodes to learn the minimum transmit powers required to successfully transmit to nearby nodes. The learning is done through RTS/CTS.

In contrast to ref. [2], [3], and our work here, ref. [4] proposed a Power Controlled Multiple Access (PCMA) protocol that uses a separate control channel for "busy tone" instead of RTS/CTS to avoid collisions, in which the signal strength of the busy tones received by a node is used to determine the power level at which this node may transmit without interfering with other on-going transmission.

Reference [7], as in our work here, does not assume the use of minimum transmit powers. It proposed to always transmit RTS/CTS and intermittently transmit data packets at the maximum power. The increased interference and EN effects due to the use of large transmit power are ignored. The approach aims to save energy, but spectral reuse is not improved.

Instead of just using minimum transmit powers, this paper proposes to adjust the transmit power of a transmitter based on its connectivity requirement with its receiver as well as potential interferences with its surrounding links. Intuitively, if the transmit power of a transmitter is decreased, it is more likely for other nodes to interfere with its receiver because of the decreased SIR; on the flip side, the interference of the transmitter to other nodes will be reduced. How to judiciously adjust the powers of neighboring nodes based on the distances among them (more exactly, the power-transfer matrix that describes their power relationships) in a distributed and parallel manner is our key focus here. As described earlier, there is generally a tradeoff between EN and HN. This paper primarily focuses on elimination of HN entirely in the network as far as algorithm design is concerned. The algorithms, however, are actually amenable to modifications that aim to decrease EN at the cost of some HN, to improve overall network throughput at the cost of unfairness in the network. This will be demonstrated in Section V of this paper.

The rest of this paper is organized as follows. Section II defines the graph models that capture the interference relationships among links to facilitate algorithmic design later. Section III presents our first power control algorithm called *Decoupled Adaptive Power Control (DAPC)*, in which each node only monitors its local surrounding to effect its own power adjustment. Different nodes can compute and adjust their transmit powers simultaneously while making sure that no new interference relationships and HN will be created in the process. Section IV presents our second power control algorithm called *Progressive-Uniformly-Scaled Power Control (PUSPC)*, which performs better than DAPC by solving a deadlock problem from which DAPC may suffer. The combination of DAPC followed by PUSPC is also investigated. Section V evaluates the performance of our proposed algorithms based on the criteria of network capacity, fairness, and amounts of EN and HN. Finally, Section VI concludes this paper.

II. GRAPH MODELS FOR CAPTURING TRANSMISSIONS CONSTRAINTS AND HIDDEN-NODE PROBLEMS

This section considers the graph models for capturing simultaneous-transmissions constraints and HN to facilitate algorithmic design. Links in the network are mapped to vertexes, and links that interfere or interact with each other are related through edges in the graph.

Subsection A provides an example to illustrate the shortcomings of power control with minimum transmit powers. Subsections B, C, and D present the components in our graph models. Specifically, subsection B considers the physical-collision constraints due to the receiver's inability to decode its signal when the powers received from other transmitting sources are large (i.e., small SIR). They form the basis of a link-interference graph. Subsection C considers the protocol-collision-prevention constraints imposed by carrier sensing of 802.11 against simultaneous transmissions. They form the basis of a protocol-collision-prevention graph. Subsection D defines the "ideal" carrier-sensing operation. They form the basis of an ideal protocol-collision-prevention graph. Subsection E defines HN and EN in a formal manner in terms of the differences between the *802.11 protocol-collision- prevention graph* in subsection C and the *ideal protocol- collision-prevention graph* in subsection D. Finally, subsection F introduces a metric for performance evaluation purposes.

A. An Example Illustrating Shortcomings of Minimum-Transmit-Power Approach

We assume the following power-transfer relationship: $P(a,b) = k \cdot P_a / r^{\alpha}$, where $P(a, b)$ is the power received by node b from node a; P_a is the transmit power of node a; r is distance between the two nodes; $\alpha > 2$ is the path-loss exponent; and k is a constant. Let T_i and R_i denote the transmitter and receiver of link i. For brevity, we also use T_i and R_i to denote their positions. So, $| a - b |$ denotes the distance between nodes a and b. We also assume that the SIR requirement, K, is such that if $KP(T_2, R_1) > P(T_1, R_1)$, then T_2 will interfere with the reception at R_1.

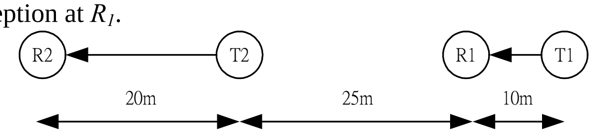


Figure 2. An example illustrating the shortcomings of minimum-transmit-power approach.

Consider links 1 and 2 in Figure 2, and assume basic-mode access with no RTS/CTS. The default parameter values used in NS-2 [8] are $K = 10$, $k = 5$, $\alpha = 4$, *TxRange* = 250m, *PCSRange* = 2.186**TxRange* = 550m, Rx_{th}=3.652e-10W, where *PCSRange* and *TxRange* are respectively the physical carrier-sensing range and transmission range, Rx_{th} is the minimum received power needed for signal detection. The corresponding transmit power given the above *TxRange* and Rx_{th} is 0.281W.

By plugging in the above NS-2 parameter values, we find that $KP(T_2, R_1) < P(T_1, R_1)$, but $KP(R_1, T_2) > P(R_2, T_2)$ according to the locations of links 1 and 2. This means that the ACK of R_1 can interfere with the reception of ACK from R_2 to T_2. However, since both T_2 and R_1 are within the *PCSRange* = 550m, the potential collision can be prevented by physical carrier sensing. So, there is no HN.

Suppose now we adjust the transmit powers of the four nodes to their minimum. After the adjustment, $P(T_1, R_1) = P(R_1, T_1) = P(T_2, R_2) = P(R_2, T_2) = Rx_{th}$, and the *TxRanges* of link 1 and link 2 become 10m and 20m respectively. Now, $KP(R_1, T_2)$ = 9.3e-11 < $Rx_{th} = P(R_2, T_2)$, but $KP(T_2, R_1)$ = 1.5e-9 > $Rx_{th} = P(T_1, R_1)$. Thus, the DATA packets of T_2 can interfere with the reception of DATA from T_1 to R_1 now. Moreover, *PCSRange* of $T_1 = 2.186 \times 10 = 21.86 < |T_2 - T_1|$ after power control. This means link 1 cannot forewarn link 2 when link 1 transmits. So, we see that the use of minimum transmit powers creates the possibility of DATA-DATA collisions. Furthermore, these collisions cannot be prevented by carrier sensing, causing the classical HN phenomenon. The use of minimum transmit powers are highly undesirable in this case.

We could also find examples in which HN is eliminated by using minimum-transmit power. However, according to our simulation results, more HN instances are created than eliminated by the minimum-transmit power approach. For example, in a randomly generated ad-hoc topology with 100 links in a domain of 1x1 km^2, there are originally 106 HN instances; but the number of HN instances increases to 542 after adopted the minimum-transmit-power approach. The reader is referred to Part E for our definition and measurement of HN in the network.

The above example points out that one must consider not just the power requirement of a link in terms of its SNR (i.e., the minimum power required at the receiver Rx_{th} so that the signal is sufficiently above the noise floor), but its SIR with respect to the potential interferences with the surrounding links. That is the basis on which the power control algorithms in this paper are designed. To aid our algorithm designs, the next few subsections describe graph models for capturing the relationships among links within vicinity of each other.

B. Link-Interference Graph from Physical-Collision Constraints

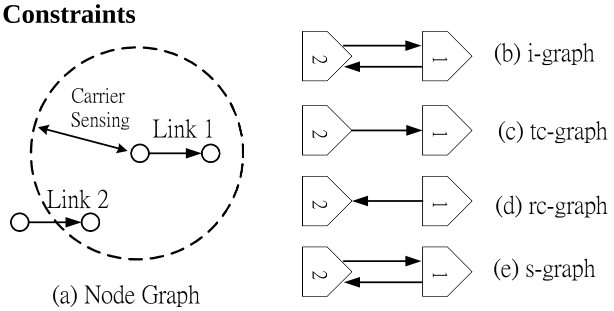


Figure 3. Mapping of a network topology a) to b) i-graph, c) tc-graph, d) rc-graph and e) s-graph.

A *Link-Interference Graph (i-graph)* can be used to represent the physical-collision constraints graphically. It basically captures the effects of SIR among links. Consider the simple network topology in Figure 3(a). As illustrated in Figure 3(b), in an i-graph, an arrow-shape vertex represents a wireless link with the arrowhead pointing toward the receiver.

In the example of Figure 3(b), there is a directional i-edge from vertex 1 to vertex 2, and vice versa, because the transmitter of link 1 and receiver of link 2 are so close to each other that they interfere with the reception at each other. DATA of T_1 may collide with DATA of T_2 at R_2; and ACK of R_2 may collide with ACK of R_1 at T_1. Although not the case in Figure 3, in general it is possible that there is an i-edge from link 2 to link 1 but not the other way round direction due to the differences in link length and powers used.

More formally, there is an i-edge from vertex 2 to vertex 1 if any of the constraints (1) – (4) is satisfied.

$$P_{T1}\left|T_2 - R_1\right|^{\alpha} < KP_{T2}\left|T_1 - R_1\right|^{\alpha} \quad \textbf{(1)}$$

$$P_{R1}\left|T_2 - T_1\right|^{\alpha} < KP_{T2}\left|T_1 - R_1\right|^{\alpha} \quad \textbf{(2)}$$

$$P_{T1}\left|R_2 - R_1\right|^{\alpha} < KP_{R2}\left|T_1 - R_1\right|^{\alpha} \quad \textbf{(3)}$$

$$P_{R1}\left|R_2 - T_1\right|^{\alpha} < KP_{R2}\left|T_1 - R_1\right|^{\alpha} \quad \textbf{(4)}$$

Constraints (1) – (4) correspond to DATA-DATA collision, DATA-ACK collision, ACK-DATA collision and ACK-ACK collision, from link 2 to link 1, respectively. Similarly, link 1 can also interfere with link 2 with four similar constraints by interchanging the indexes 1 and 2 in (1) – (4).

Note that in this paper we model links as directional links. So, bidirectional links between two nodes will be considered as two links. For example, if link 1' is the reverse-directional link of link 1, then $T_{1'} = R_1$ and $R_{1'} = T_1$ in our model, and the links will be modeled separately as vertexes 1 and 1' in our graph model.

C. Protocol-Collision-Prevention Graphs

We next consider the effect of 802.11 carrier sensing. The goal of carrier sensing is to prevent simultaneous transmissions that will collide. Two protocol-collision-prevention graphs can be used to model the carrier sensing: the tc-graph models the effect of carrier sensing by transmitters, and the rc-graph models that by receivers.

In the tc-graph, there is a directional tc-edge from vertex 1 to vertex 2 if T_2 can sense the transmission on link 1 so that if T_1 is already transmitting its DATA, T_2 will not transmit. Formally, there is a tc-edge from vertex 1 to vertex 2 if any of the inequalities (5) – (7) is true.

$$\left|T_2 - T_1\right| < VCSRange(P_{T1}) \quad \textbf{(5)}$$

$$\left|T_2 - R_1\right| < VCSRange(P_{R1}) \quad \textbf{(6)}$$

$$\left|T_2 - T_1\right| < PCSRange(P_{T1}) \quad \textbf{(7)}$$

where $VCSRange(P_a)$ is the virtual carrier-sensing range due to the transmissions of RTS/CTS by node a with transmit power P_a ; and $PCSRange(P_a)$ is the physical carrier-sensing range due to the DATA transmission by node a.

In the example of Figure 3(c), we assume T_1 and T_2 are sufficiently far apart that they cannot physically sense each other. However, T_1 can sense the CTS of R_2, but T_2 is so far away from T_1 and R_1 that it cannot sense the RTS and CTS from them. So, there is a tc-edge from link 2 to link 1 but not the other way round.

In the rc-graph, there is a directional rc-edge from vertex 1 to vertex 2 if R_2 can sense the transmission on link 1. Specifically, there is an rc-edge from link 1 to link 2 if any of the inequalities (8) – (9) is satisfied.

$$\left|R_2 - T_1\right| < VCSRange(P_{T1}) \quad \textbf{(8)}$$

$$\left|R_2 - R_1\right| < VCSRange(P_{R1}) \quad \textbf{(9)}$$

$$\left|R_2 - T_1\right| < PCSRange(P_{T1}) \quad \textbf{(10)}$$

When T_1 is already transmitting, T_2 can still transmit if there is an rc-edge, but no tc-edge, from vertex 1 to vertex 2. However, R_2 will ignore the DATA (RTS) frame and not return an ACK (CTS). The rationale for R_2 not returning an ACK (or CTS) to T_2 is that the ACK (CTS) may interfere with the ongoing transmission on link 1.

In the example of Figure 3(d), there is an rc-edge from link 1 to link 2 but not in the other direction since R_2 can sense the RTS of T_1, but R_1 is so far away from link 2 that it cannot sense any RTS/CTS from it.

D. Ideal Protocol-Collision-Prevention Graph

Subsection C was about the tc-graph and rc-graph that models the carrier-sensing operation of 802.11. However, 802.11 carrier sensing may not be ideal in that it may (i) prevent non-collision-causing simultaneous transmissions, and (ii) fail to prevent collision-causing simultaneous transmissions. We introduce the concept of a s-graph with *s-edges* (*should-forewarn edges*) to describe the *ideal* carrier-sensing operation. The comparison of s-graph, tc-graph, and rc-graph allows us to define HN and EN in a formal manner, which aids our algorithmic design later.

In an s-graph, there is an s-edge from vertex 1 to vertex 2 if link 1 should forewarn link 2 when it transmits, due to the presence of an i-edge from 1 to 2, or an i-edge from 2 to 1. Equivalently, there are two s-edges, one from 1 to 2, and one from 2 to 1, if there is an i-edge from vertex 1 to vertex 2. The definition of s-edge is as such because no matter link 1 or link 2 transmits first, transmission at link 2 will fail. Therefore, when link 2 transmits, it should first forewarn link 1 not to transmit. Similarly, when link 1 transmits, it should forewarn link 2 not to transmit. In short, s-edges always exist in pairs.

In the example of Figure 3(e), there are two s-edges between links 1 and 2, one in each direction. It turns out that there are also two i-edges in this example. However, even if there were only one i-edge between links 1 and 2, we would still have the two s-edges in both directions.

E. Definition of HN and EN and their Investigation using Graph Model

Figure 4 shows the Venn Diagram depicting the relationships among different types of edges and the inequalities that define them. In the Venn Diagram, the set elements are link duples (i, j). Each link duple (i, j) represents the relationship from vertex i to vertex j. It could be a tc-edge, rc-edge, s-edge, none of them, or a combination of them

We now provide formal definitions for HN and EN. Based on the definition, we introduce the metric to measure the severity of HN and EN in the network, which will be used in this paper to analyze our simulation results. Strictly speaking, the HN and EN phenomena are due to relationships between links rather than between nodes. However, we will continue to use these terms since they are already commonly used by others.

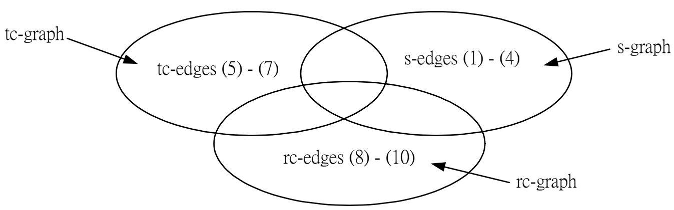


Figure 4. Relationships among s-edges, tc-edges and rc-edges in s-graph, tc-graph and rc-graph; and the constraints associated with the edges.

Definition of HN

There is HN from link i to link j if (i, j) is not a tc-edge, but is an s-edge or rc-edge. Link i is said to be hidden from link j in this case.

Definition of EN

There is EN from link i to link j if (i, j) is not an s-edge, but is a tc-edge or rc-edge. Link j is said to be exposed to link i in this case.

With respect to the above HN definition, simultaneous transmissions on links i and j cannot both be successful. However, link j cannot be prevented from transmitting when link i is already transmitting. As for the EN definition, there is actually no physical interference between links i and j in terms of their SIRs. However, the existence of a tc-edge from i to j will prevent j from transmitting when i is already transmitting; while the existence of an rc-edge will prevent the success of the transmission by link j because R_j will not reply to T_j. Thus, HN is a phenomenon whereby colliding transmissions fail to be prevented by carrier sensing, while EN is a phenomenon whereby non-colliding transmissions, or their success, are prevented by carrier sensing. Fundamentally, HN and EN are both caused by the discrepancies among s-edges, tc-edges and rc-edges.

Let us denote the set of s-edges by S, the set of tc-edges by TC, and the set of rc-edges by RC. As measures of the severity of HN and EN in the overall network, we can look at

# of HN-causing edges: $N_{HN} = |S \cup RC| - |TC \cap (S \cup RC)|$

# of EN-causing edges: $N_{EN} = |TC \cup RC| - |(TC \cup RC) \cap S|$

For the network to be HN-free, we require $TC = S \cup RC$ so that $N_{HN} = 0$. For the network to be EN-free, we require $S = TC \cup RC$ so that $N_{EN} = 0$. In general, 802.11 networks cannot be both HN-free and EN-free. We may define normalized

Miss Ratio = $N_{HN} / |S \cup RC|$

False-alarm Ratio = $N_{EN} / |S \cup RC|$

to measure the severities of HN and EN in a given network. The reason for using the normalization factor $|S \cup RC|$ is that it corresponds to the number of cases where simultaneous transmissions are not allowed, or will not be successful.

In our simulations in Section V, the network has no HN initially when our power-control algorithms start running, and the algorithms are required to maintain the HN-free property throughout their execution. Generally speaking, to maintain the HN-free property, (i) the carrier-sensing range must be sufficiently large and (ii) a so-called receiver restart (RS) mode must be effected (the reader is referred to [5] for details). As far as our work here is concerned, the RS mode is assumed and we simulate the network in basic mode with the initial physical carrier-sensing range, *PCSRange* set to 3.78 *TxRange*, the maximum transmission range of DATA at the initial transmit power.

F. Attacking Cases

This subsection introduces another performance metric, *number of attacking cases*, which corresponds to the number of cases where simultaneous transmissions are either *not* allowed, or where allowed, will not be successful. Link i is said to be attacking link j if (i, j) is an i-edge, a tc-edge, or a rc-edge.

The number of attacking cases in a network is the sum of the number of attacking cases from link i to link j over all i and j. Specifically, for all (i, j), $i \neq j$:

If (i, j) is an i-edge, then add 2 to # attacking cases;
else if (i, j) is a tc-edge, then add 1 to # attacking cases;
else if (i, j) is an rc-edge, then add 1 to # attacking cases.

The above enumeration process takes into account the order of transmissions. If (i, j) is an i-edge, it does not matter whether i or j transmits first, signal at j will be corrupted. So, there are two cases where i can "attack" j. On the other hand, if (i, j) is a tc- or rc-edge, transmission at link j will not be allowed or will fail only if link i transmits first. So, there is only one case. If there are L directional links, then the above summation will be over $L(L-1)$ link pairs.

III. DECOUPLED ADAPTIVE POWER CONTROL (DAPC)

This section presents our first distributed adaptive power control algorithm, *Decoupled Adaptive Power Control (DAPC)*. The main essence of DAPC is to decouple the power adjustments of individual links to the extent that is possible, so that many links can adjust their powers simultaneously in a distributed and parallel manner. In DAPC, (i) a node only needs to gather information from nearby nodes that are within a certain "radius" to it; and (ii) powers used by links that are far apart can be adjusted simultaneously. Each individual node will perform its power adjustments based on its own computation through a number of iterations. In Subsection A, we first discuss how much power can be adjusted by a link in each iteration in our algorithm. Subsection B provides a Power Exchange Algorithm (PE) [9] for links to gather power information and discuss how the "radius" in (i) can be bounded with the concept of *Interaction Range* (*IntRange*). Based on the principles in Subsections A and B, Subsection C discusses the implementation of DAPC, which guarantees that no new i-edges or HN will be created in each iteration. Subsection D points out, however, that DAPC may face a deadlock problem that limits its performance. Deadlock-free power control will be presented in Section IV.

A. Per-iteration Power Adjustment

In this algorithm, when links perform power adjustment in each iteration, they assume the transmit powers of its surrounding links remain unchanged. The link must make sure that 1) the connectivity between its transmitter and receiver can be maintained; 2) the power reduction does not create new i-edges from other links to itself, even assuming other links do not change their powers; and 3) the carrier-sensing range with the reduced power is still sufficient to cover interfering nodes, such that no new hidden nodes are created. Note that if all links perform 2), no new i-edges will be created in the network because each link assumes the worst-case SIR in its power adjustment. The steps are elaborated below for an arbitrary link labeled as link 1.

1. **Ensuring reduced power satisfies minimum received power threshold to maintain link connectivity:** To guarantee connectivity from T_1 to R_1, the minimum transmit power of T_1 must be bounded below by

$$P_{\min}(T_1) = \frac{Rx_{th}}{G(T_1, R_1)} = \frac{P_{T1}}{P(T_1, R_1)} \times Rx_{th} \qquad \textbf{(11)}$$

where Rx_{th} is the minimum necessary received signal strength, and $G(i, j) = P(i, j)/P_i$ is the power-gain function from node i to node j that can be computed from the current transmit power P_i used by node i and the current power of node i received by node j, $P(i, j)$. Similarly, the minimum transmit power of R_1 must be bounded below by

$$P_{\min}(R_1) = \frac{Rx_{th}}{G(R_1, T_1)} = \frac{P_{R1}}{P(R_1, T_1)} \times Rx_{th} \qquad \textbf{(12)}$$

$G(T_1, R_1) = G(R_1, T_1)$ can be found from the Power Exchange Algorithm described in Subsection B.

2. **Ensuring reduced power does not create new i-edges:** To ensure that *no new i-edges to vertex 1 will be created when T_1 and R_1 reduce their transmit powers*, we need to consider the interferences from surrounding links. Let N_{T1} and N_{R1} be respectively the sets of nearby transmitting and receiving nodes that are not currently interfering with T_1 and R_1, but which may potentially do so if the powers of T_1 and R_1 are reduced too aggressive. As a conservative measure, we assume the powers of the nodes in N_{R1} and N_{T1} are not changed when computing the acceptable new powers of T_1 and R_1. We require

$$P_{adjusted}(T_1) \;\geq\; KP(n, R_1)/\mathrm{G}(T_1, R_1) \quad \forall n \in N_{R1} \qquad \textbf{(13)}$$

$$P_{adjusted}(R_1) \;\geq\; KP(n, T_1)/\mathrm{G}(T_1, R_1) \quad \forall n \in N_{T1} \qquad \textbf{(14)}$$

Note that (13) is to ensure there is sufficient SIR at R_1 for the DATA on link 1, and (14) is to ensure there is sufficient SIR at T_1 for the ACK on link 1. In general, N_{T1} and N_{R1} do not need to cover all nodes in the network. Only nodes n that satisfy the following need to be considered:

(i) $n \in N_{T1}$ if and only if $P(n, T_1) \;\geq\; Rx_{th} / K$

(ii) $n \in N_{R1}$ if and only if $P(n, R_1) \;\geq\; Rx_{th} / K$

3. **Ensuring *PCSRange* of reduced power is enough to cover interfering nodes:** In this paper, we focus on the basic-access mode. This requirement is to ensure that the physical carrier sensing in 802.11 continues to avoid HN after each power adjustment. Let M_{T1} denote the set of transmitters whose link has an s-edge to link 1 and vice versa. This means that $\forall\, m \in M_{T1}$, the *PCSRange* of T_1 must be able to reach m. Note that the difference between M_{T1} and N_{T1} is that the former refers to nodes whose links already have interference relationships with the link 1, whereas the latter refers to nodes that do not currently interfere with T_1 but may do so if power adjustment is not done right. Before T_1 transmits, it must be able to warn the nodes in M_{T1} not to transmit through physical carrier sensing. Therefore, to maintain the HN-free property, the following inequality must be satisfied:

$$P_{adjusted}(T_1) \;\geq\; Rx_{th}^{PCS} \;/\; G(T_1,\; m) \quad \forall m \in M_{T1} \qquad \textbf{(15)}$$

where Rx_{th}^{PCS} is the receiver sensitivity threshold for PHY header, which is generally smaller than Rx_{th} so that *PCSRange* is larger than *TxRange*. This is achieved by having the PHY header transmitted at a lower rate than the DATA payload. For example, if the ratio of the transmission rate of the DATA payload to the transmission rate of the PHY header is r_{PCS}, as an approximation, we have $Rx_{th}^{PCS} = Rx_{th} \,/\, r_{PCS}$. Note that we have assumed the same transmit power is used to carry DATA/ACK and PHY on a link.

In DAPC, the transmitter and receiver of a link may use different power levels. Steps 1, 2, and 3 are combined as follows. We set $P_{adjusted}(T_1)$ to the maximum of (11), (13) and (15). Then, we set $P_{adjusted}(R_1)$ to the maximum of (12) and (14).

The nodes in N_{T1}, N_{R1}, and M_{T1} in steps 2 and 3 define an *Interaction Range* (*IntRange*) over which other links can interfere with or can potentially interfere with link 1. Specifically, faraway nodes outside of *IntRange* not belonging to N_{T1}, N_{R1}, and M_{T1} need not be considered by link 1 when it adjusts the transmit powers used by its transmitter for DATA and its receiver for ACK. Note that not all links within *IntRange* can interfere with link 1, but all links outside *IntRange* are guaranteed not to do so.

B. Power Exchange Algorithm

In ref. [9], a Power Exchange Algorithm (PE) has been proposed for establishing the i-graph of a network. Our power adjustment procedure in Subsection A requires not only the knowledge of the current i-edges, but also the power-transfer relationships between nearby nodes so that we can ensure that no new i-edges will be created and the *PCSRange* is sufficient after power adjustment. We extend the PE in [9] for our purpose here.

Power-Exchange packets (PE packets) are special packets periodically broadcasted by nodes to exchange power information with neighbors. We assume the transmit powers of these packets are the same as the transmit powers of regular packets like DATA/ACK/RTS/CTS.

Consider an arbitrary node a. The PE packets sent by node a contain three types of information: (i) Active links (a, b) or (b, a), where b is any other node which forms an active link with a; (ii) Transmit power P_a of node a; (If node a is an AP, we assume it uses different P_a for different client stations and establishes multiple links with clients) (iii) "Power set", as described below. The identity of the node a is implicit in the MAC address of its PE packets.

- Node a monitors the power it receives from other nodes and keep this information in a power set $PS_a = \{[b, P(b, a)], [c, P(c, a)], \ldots\}$.
- Node a periodically broadcasts a PE packet at a rate lower than the data rate to increase the transmission range.
- Node a gathers information from the PE packets received from its neighbors by measuring the powers of the received PE packets as well as looking into their contents.

Condition for Correct Operation of PE: Interaction Range

The following conditions are sufficient to ensure that the necessary information, including the power-transfer relationships required for the computation in Subsections A can be gathered by the PE algorithm:

$$Rx_{th}^{PE} < Rx_{th} \,/\, K \qquad \textbf{(16)}$$

$$Rx_{th}^{PE} \leq Rx_{th}^{PCS} \qquad \textbf{(17)}$$

where Rx_{th}^{PE} is the receiver sensitivity threshold for PE packets, it should be set as the maximum of (16) and (17). The purpose of (16) and (17) is to ensure that the PE packets sent out by other interfering links or potentially interfering links within the *IntRange* can be received, so that the required information needed to execute steps 2 and 3 in Subsection A can be obtained from the PE packets.

Due to the space limit here, the reader is referred to [1] for the formal proof of the above conditions for the correct operation of PE Algorithm.

C. Implementation of DAPC

We now discuss implementation issues with regard to the PE algorithm. According to (16) and (17), the receiver sensitivity for PE packets must be higher than that for regular DATA packets and at least the same as that for PHY header. If each node transmits its PE and DATA packets at the same power (but different nodes may still use different powers), then for that node, we could transmit PE packets at the same rate as that of PHY header.

Consider IEEE 802.11b. The data rate of DATA is 11Mbps. The PHY header is transmitted at 1Mbps. Inequality (17) can be satisfied if we also transmit PE packets at 1 Mbps. In addition, to the extent that the receiver sensitivity can be improved by a factor of 11 (to maintain the same energy per bit for 1Mbps and 11Mbps), then (16) can also be satisfied. This is because $Rx_{th}^{PE} = Rx_{th}^{PCS} = Rx_{th} / 11 < Rx_{th} / K$ with K=10 (the typical 10dB SIR requirement).

The above argument is based on the same-energy-per-bit assumption. Since different coding schemes are used for 1Mbps and 11Mbps, this assumption may not apply strictly. NS-2 simulates what is found in a commercial product. In its default setting [8], *PCSRange* = 550m, while *TxRange* = 250m. So, *PCSRange* = 2.2**TxRange*. Assuming PE packets has the range as the PHY header, *PCSRange*, the implied receiver sensitivities are related by $Rx_{th}^{PE} / Rx_{th} = 1/2.2^{\alpha}$, where α is the path-loss exponent. Any $\alpha > 2.92$ guarantees that $Rx_{th}^{PE} / Rx_{th} < 1/K$ where K = 10. The default α value in NS-2 is 4. So, PE data rate of 1Mbps is also sufficient by this argument.

One may notice that there are potential collisions of broadcast PE packets with other packets, including regular packets and other PE packets. Since broadcast packets have no ACK, each PE packet may need to be transmitted several times (e.g. three times in our assumption) to make sure it is received by all the intended receivers.

The distributed DAPC, however, is robust in that even if all the three PE packets were missed by an intended receiver node, the node will simply assume the "worst-case" in which it is assumed that the node whose PE packets have been missed continue to use the previous higher power.

D. Deadlock Problem in DAPC

In DAPC, every link adjusts its power while assuming the powers of neighboring links remain unchanged. In this case, they may run into a deadlock. An illustrating example is shown in Figure 5. In the figure, links 1 and 2 are of unit length. Suppose that all the nodes are currently using the same transmit powers. The nodes of links 1 and 2 do not interfere with each other according to inequalities (1) – (4), since they are separated by distance of $K^{1/\alpha}$ units, which is just equal to the interference margin.

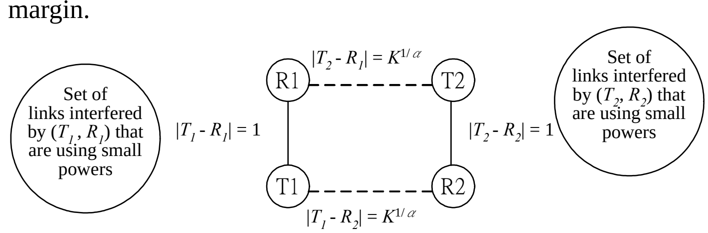


Figure 5. Illustration of the deadlock problem of DAPC.

However, if link 1 (link 2) adjusts its power down while link 2 (link 1) does not adjust its power, a new i-edge will be created from link 2 to link 1. Thus, according to DAPC, no power reductions will be allowed for links 1 and 2.

On the other hand, if both links 1 and 2 adjust their powers down by the same amount, i-edges will not be created between them since the SIR remains the same. Reducing powers as such may be desirable because it may reduce the interferences of links 1 and 2 to other nearby links, leading to elimination of i-edges from links 1 and 2 to them. DAPC cannot achieve this, and will be stuck in the suboptimal solution in which the powers of links 1 and 2 will remain high indefinitely. We refer to this as the *deadlock problem.*

Definition of Deadlock: An algorithm is said to run into a deadlock if:

(1) no further power adjustment is possible according to the algorithm;

(2) however, one can identify a set of links whose powers can be further adjusted down simultaneously without creating new i-edges while maintaining link connectivity and HN-free property.

Note that part (2) of the above definition only requires that no new i-edges are created. In general, reducing powers may also eliminate some of the old i-edges, although this is not a requirement according to the definition. The idea is that we would like to use as small powers as possible.

To reduce the likelihood of deadlock (or more specifically, to ensure that when deadlocks occur, the power levels are already low), we may perform Uniformly-Scaled Power Control (USPC) [1], where we selects a *common* and *uniform* initial transmit power for all nodes that is sufficiently low before launching DAPC. Figure 6 shows the performance of DAPC and DAPC with pre-USPC. The reader is referred to Section V for the detailed simulation settings. For the case with pre-USPC, we set the initial transmit power so that the initial *TxRange* corresponds to half the diagonal of a square in the grid.

We see from the figure that DAPC with pre-USPC (DAPC-PU) can achieve a smaller number of attacking cases than DAPC alone. In particular, the performance of DAPC-PU is within 22% from the benchmark, which corresponds to the result of the minimum-transmit power approach, in which all nodes use just enough power to maintain its link connectivity. Note that the benchmark case is one in which there are the fewest numbers of tc- and rc-edges, but in which there may be excessive numbers of HN.

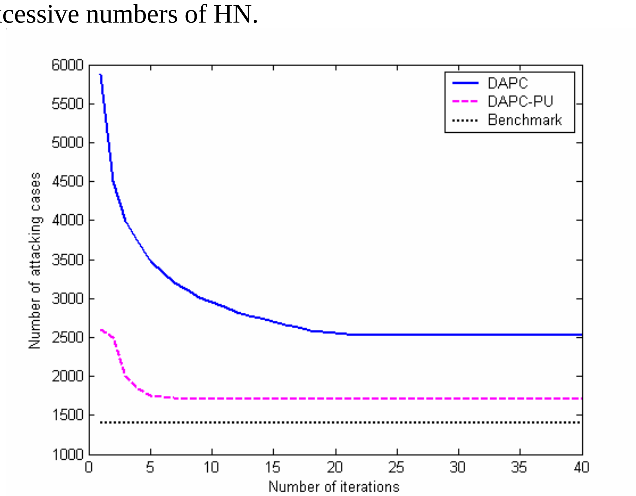


Figure 6. Simulation results of DAPC and DAPC-PU.

IV. PROGRESSIVE-UNIFORMLY-SCALED POWER CONTROL (PUSPC): DEADLOCK-FREE DESIGN

This section presents our second distributed adaptive power control algorithm, called *Progressive-Uniformly-Scaled Power Control (PUSPC)*, which is deadlock-free. In Subsection A, we present the details of the algorithm, followed by the proof of its deadlock-free property in Subsection B. Subsection C considers deadlock-free resolution for DAPC by applying the concept of PUSPC.

A. Algorithm of PUSPC

In PUSPC, we divide the links into *PowerControlSet* and *FinishedSet*. Initially, all links are in the *PowerControlSet* and they will start with the same initial power. They then reduce their powers by a common quantized size in each iteration and the algorithm is synchronized. As time progresses, some links will be placed in the *FinishedSet* and their powers will not be further adjusted while the links in *PowerControlSet* continue to reduce their powers in future iterations. At any one time, all links in *PowerControlSet* have the same uniform powers, while links in *FinishedSet* may have different powers.

In each iteration, each node *k* whose link is in *PowerControlSet* can further reduce its power by the quantized size if three conditions below are satisfied:

(i) Its link will not be disconnected after power adjustment.

(ii) No new i-edge will be formed from links in *FinishedSet* to its links after power adjustment. Note that no *new* i-edge will be formed among links in *PowerControlSet* since their powers are the same and adjusted by the same amount – i.e., there is no change in SIR. In addition, no new tc- or rc-edges will be created by reducing power.

(iii) Its *PCSRange* is enough to cover interfering nodes after power adjustment. In this constraint, node *k* needs to check that the *PCSRange* is still sufficient to reach the nodes in interfering links.

Note that (i), (ii) and (iii) are similar to steps 1, 2 and 3 in Section III.A, except that for (ii), we assume the other links in *PowerControlSet* adjust their powers by the same amount in the same iteration; whereas in step 2, we assume the other links will use the powers that they used in the previous iteration. Essentially for PUSPC, N_{RI} and N_{TI} in (13) and (14) should include only nodes whose links are in *FinishedSet*.

In PUSPC, we can add one more bit in the PE packet to indicate whether a link is in *PowerControlSet* or *FinishedSet*, and similar to DAPC, each node only needs to monitor the PE packets from neighboring nodes within its *IntRange.*

The quantized step size for power reduction is a crucial factor that affects the efficiency and accuracy of PUSPC. It is a trivial fact that links can reach a smaller power level with a smaller step size or more iterations, and a larger number of attacking cases can be reduced. In our simulations in this paper, we adopt the dB step size, i.e., we reduce the power by a constant amount (in unit of dB) in each iteration. The simulation result of PUSPC with the step size of 1dB is shown in Figure 9.

B. Deadlock-free Property of PUSPC

We now prove that PUSPC is deadlock-free. As the power of nodes in *PowerControlSet* is adjusted down in successive iterations, there comes an iteration when the power of a "critical" link cannot be adjusted further, and this critical link will then be placed in the *FinishedSet*. There are two possible reasons why the power of the critical link cannot be adjusted further:

(i) Reducing the power further may cause either the critical link to be disconnected, or may cause its carrier-sensing range to fail to cover an interfering link from *FinishedSet* for HN-free operation.

(ii) Reducing the power further may create new i-edges from some link in *FinishedSet* to the critical link.

Note that (i) is not a cause of deadlock, because the power of the critical link cannot be adjusted down if the link connectivity and HN-free requirements are to be preserved. (see definition of deadlock condition (2) in Section III.D). That leaves us to prove that (ii) will not cause deadlock either. Since we are using quantized step size here, we redefine part of the definition of deadlock (2) in Section III.D to "(2) however, one can identify a set of links whose powers can be further adjusted down by the quantized step size, *S*, simultaneously without creating new i-edges while maintaining link connectivity and HN-free property." With this modification of deadlock definition, we have the following proposition:

Proposition 1:
There is no deadlock in the solution produced by PUSPC.

Proof: Consider a link, say link 1, in *PowerControlSet*. Suppose that in the current iteration, link 1 is the "critical" link that would violate constraint (ii) if its power were reduced further, and that there would be a new i-edge formed from link 2 in *FinishedSet* to link 1. In PUSPC, Link 1 will be moved to *FinishedSet*, and the power adjustment will be the power level just above the critical power adjustment. We show that deadlock involving link 1 and other links in *FinishedSet* is not possible at the end of this iteration. By induction, after the last iteration when all links have been added to *FinishedSet*, there will be no deadlock in the overall network.

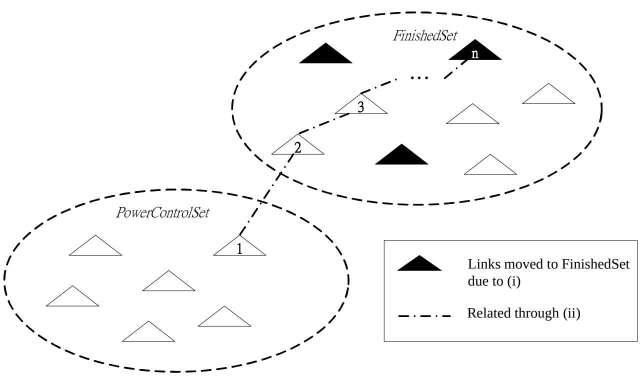


Figure 7. Graphical illustration of deadlock-free in PUSPC.

Suppose we assume on the contrary that there is deadlock involving link 1 and some other links in *FinishedSet*. That means it is possible to adjust the power of link 1 plus the power of some other links in *FinishedSet* without creating new i-edges. Say, the additional power adjustment is $\delta p > S$. If the power of link 1 is reduced by this amount, the power of link 2 should also be reduced by the same amount to maintain the same SIR so that no new i-edge is formed from link 2 to link 1. By the same token, if the power of link 2 is reduced by δp, the power of another link, say link 3, must also be adjusted since link 2 was a critical link in a previous iteration. Continuing this argument allows us to identify a set of links – link1, link 2, …, link n – whose powers must be adjusted down together, or else new i-edges may be formed. At some point, we will find a link, say link *n*, whose power cannot be adjusted down because of (i) rather than (ii) – in the "worst case", all links in *FinishedSet* are identified; however, the first link included in *FinishedSet* in the first iteration is always due to (i). A graphical illustration is shown in Figure 7, where the black triangle represents the link whose power cannot be adjusted because of (i). Certainly, it is not possible to adjust the power of link *n* by δp. We have thus shown that it is not possible to simultaneously adjust the powers of links 1, 2, ..., *n* simultaneously without creating an i-edge while maintaining link connectivity and HN-free property.

C. Deadlock Resolution for DAPC using PUSPC

As discussed in Section III, DAPC is not deadlock-free. Some nodes may reach a deadlock and remain at a high power level. As a result, some reducible tc- and rc-edges become irreducible. Although DAPC-PU discussed in Subsection III.D can achieve a smaller number of attacking cases, it is not deadlock-free. Also, it may be inconvenient to have to conduct the pre-USPC phase, since an implicit assumption is that we can find a "low" common initial power that all nodes can adopts.

For deadlock-free designs, an alternative to PUSPC in the previous subsection is to modify DAPC to eliminate deadlocks. A two-step approach which consists of DAPC followed by PUSPC for deadlock resolution can be used:

1) *Execute DAPC and identify the links that have reached their maximum reducible powers*: After the DAPC algorithm, we want to execute PUSPC next. Let us put all links in *FinishedSet* first and then identify a set of links in *FinishedSe*t that can be moved back to *PowerControlSet* for further power reduction. There is a subset of links in *FinishedSet* whose powers cannot be reduced further because of condition (i) in Subsection B. For illustration, these links are represented by black triangles in Figure 8. There are also other links represented by white triangles in Figure 8. These links cannot reduce their powers because of (ii) in Subsection B. So, we may form many "trees". Some trees have black triangles as roots and some not. The links in those trees with black roots cannot reduce their powers further (see proof of Proposition 1).

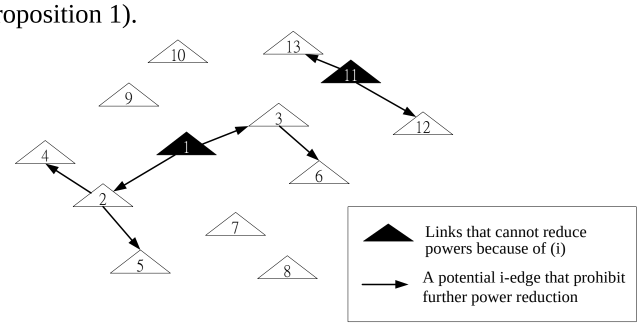


Figure 8. Graphical illustration of trees formed from links that have reached their maximum reducible powers.

2) *Perform PUSPC on trees with no black triangles*: The links in trees with black triangles remain in the *FinishedSet.* The remaining links (links 7, 8, 9 and 10 in the example in Figure 8) are put into *PowerControlSet*. Links in *PowerControlSet* are actually deadlock-causing links in DAPC. Their powers can be reduced together to break the deadlocks. With these two sets, we may then perform PUSPC as described in previous subsection. Note that unlike in the original PUSPC, where all nodes in *PowerControlSet* have the same power, in the post-DAPC PUSPC here, the nodes in *PowerControlSet* may have different powers. However, this does not cause any problem if we reduce the power using dB step size, so that SIRs among nodes in *PowerControlSet* remain the same if their powers are adjusted by the same dB amount.

Figure 9 shows the performance of DAPC with this Deadlock Resolution (DAPC-DR) and the original PUSPC in the previous subsection, both at a step size of 1dB. In the figure, the second portion of the curve for DAPC shows the performance of the deadlock resolution. Originally, DAPC stops with 2521 attacking cases, the deadlock resolution further pulls down the number of attacking cases to 2233. Although in this simulation setting, the number of attacking cases reduced by deadlock resolution is not large, it is still important to have it to guarantee the performance of DAPC. In other settings, there could be two links that hold on to large transmit powers because of deadlock. Such links with large powers will cause a large number of tc- and rc-edges in their neighborhood, resulting in unacceptable performance at portions of the network within their vicinities.

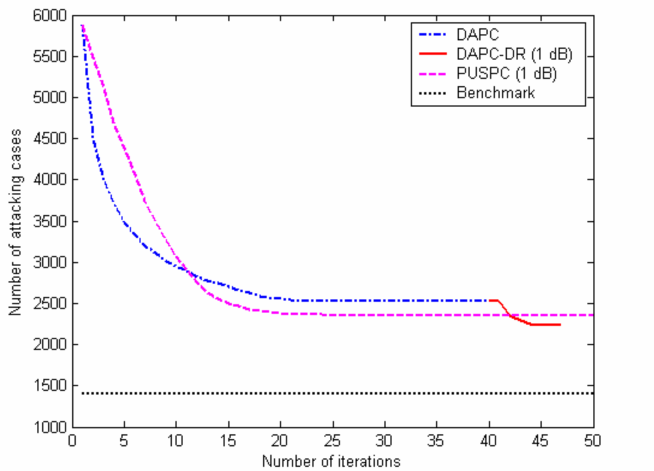


Figure 9. Simulation results of PUSPC and DAPC-DR.

V. NUMERICAL PERFORMANCE RESULTS

In our simulations, we use a grid topology in a 1x1 km^2 domain. Each square in the grid contains an AP at the center. There are 25 APs, and 100 client stations are placed randomly in the whole domain. Each client is connected to its closest AP. The initial transmit power of all nodes is 281.8mW. We simulate the network in the basic-access mode with the initial *PCSRange* = 3.78 x *TxRange*, and Receiver Restart (RS) Mode is turned on. This setting guarantees that *the initial network is HN-free.*

Table I summarizes the overall simulation results of ordinary non-power-controlled 802.11 (with a constant power level of 281.8mW), minimum-transmit-power approach, DAPC-DR (step size = 1dB) and PUSPC (step size = 1dB). The network capacities in Table I are obtained by simulations in NS-2 [8]. A UDP link is established from each client to its associated AP. Data rate of 11 Mbps, packet size of 1460 bytes are assumed.

Table I. Comparison of DAPC-DR, PUSPC with the minimum-transmit-power approach and ordinary non-power-controlled 802.11.

	802.11 with RS Mode (HN-free)	Min-pow	DAPC-DR (1dB)	PUSPC (1 dB)
# attacking cases	5879	1406	2233	2335
Total Network Capacity (NS-2) (Mbps)	19.69	46.63	45.09	49.00
# HN-causing edges	0	386	0	0
Miss Ratio (%)	0	45.31	0	0
# EN-causing edges	4428	80	1178	977
False-alarm Ratio (%)	519.72	9.39	198.99	129.58
Jain's Fairness Index	0.4	0.3	0.41	0.39

In general, Min-pow, DAPC-DR and PUSPC have comparable capacities that are at least two times that of the ordinary non-power-controlled 802.11 network. PUSPC has the highest capacity (5% higher than Min-pow). However, DAPC-DR and PUSPC are HN-free while Min-pow is not.

To demonstrate that unfair bandwidth distribution among links can be caused by the presence of HN, we employ the Jain's Fairness Index [10] to measure the fairness of the networks. In Table I, we can see that the Jain's Fairness Index for DAPC-DR and PUSPC is about 30% larger than that of Min-pow, which shows that DAPC-DR and PUSPC in general can achieve fairer network bandwidth distribution among links than Min-pow, thanks to the elimination of HN.

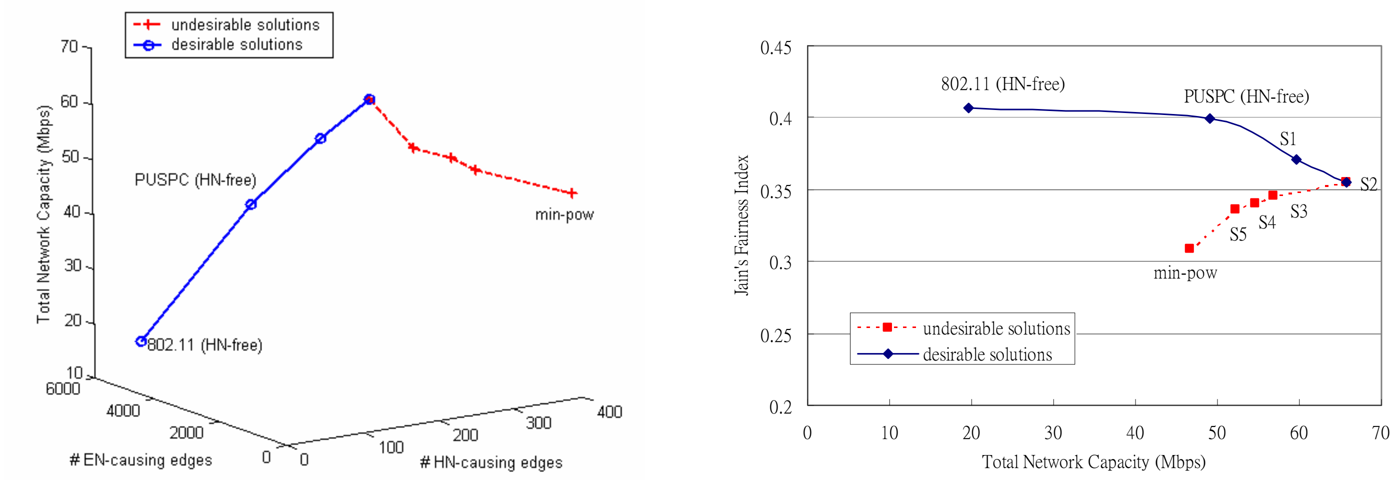
Trade-off between EN and HN

We have so far focused on algorithms that eliminate of HN entirely in the network. The HN-free algorithms we have considered, however, are actually amenable to modifications that aim to decrease EN at the cost of some HN. Figure 10a shows the variation in network capacity with the trade-off between HN and EN. The settings considered include 1) a hidden-node free network without power control at the constant power level of 281.8mW; 2) a hidden-node free network with PUSPC; 3) a range of networks with constraint (iii) of PUSPC relaxed to allow progressively increasing number of HN-causing edges; and 4) the minimum-transmit-power approach. We also map the plot in Figure 10a point-by-point to a Jain's Fairness Index versus total network capacity plot, as shown in Figure 10b.

The relaxation in 3) above is achieved by means of disregarding the coverage requirement of *PCSRange*. More specifically, we allow inequality (15) to be violated for a maximum of *d* times for each link. Thus, we can tune the degree of HN in the network by tuning *d*. Each time (15) is violated, the # HN-causing edges increases while the # EN-causing edges may decrease. As *d* increases, the curve asymptotically approaches the case with minimum-transmit powers. As a reference, we also plotted the throughput of Min-pow as the last point of the curve.

From Figures 10, we observe that for the solid-line part of the curve the total network capacity increases while we increase HN, validating the trade-off between throughput and fairness. In the solid-line region, the throughput cannot be improved without sacrificing fairness. In this region, we cannot definitely say one solution is more effective than another. The design decision is pretty much an exercise in finding the right balance between throughput and fairness, the trade-off of which is caused by the trade-off of the degrees of EN and HN in the network.

With reference to Figure 10 again, note that there is a turning point beyond which further increasing HN, although can reduce EN, actually causes the network capacity to go down. This corresponds to the dotted-line part of the curve, which is an undesirable operating region. The reason that the throughput drops in this region could be that when there are too many hidden nodes in the network, the carrier-sensing mechanism fails to prevent a large number of collisions. In the extreme that the network operates without carrier sensing, we are essentially left with an Aloha network, whose throughput is well known to be quite a bit lower than a network with carrier sensing in an analysis in which spectrum spatial re-use is not considered.



a) Total throughput against #HN and #EN-causing edges

b) Jain's Fairness Index against total throughput

Figure 10. Total network capacity and fairness with the trade-off between HN and EN from HN-free scenario to the minimum-transmit-power approach.

VI. CONCLUSION

Spectral reuse in wireless networks can be optimized by judicious transmit-power control. Most previous investigations [2] [3] adopt the minimum-transmit-power approach which aims to reduce EN. HN and its associated problems remain. Our investigation has been an attempt to find better schemes than the minimum-transmit-power approach. Overall, the main contributions of this paper are three-fold:

1. We have shown by an example and simulation results that power control with minimum transmit powers can create HN, which may cause a number of performance problems, including unfair bandwidth distributions in the network.
2. We have proposed and investigated two distributed adaptive power control algorithms: DAPC and PUSPC. When adjusting powers, these algorithms make sure that (i) no new interference relationships will be created beyond those already in existence; and (ii) no new hidden nodes will be created. These algorithms can achieve high spectral reuse by reducing EN while avoiding HN entirely. Thus, increasing the system throughput while alleviating the unfair bandwidth-distribution problem. In addition, these algorithms can be modified to allow some degree of HN for further reduction of EN to increase throughput at the expense of fairness.
3. We have showed that the *desirable* operating points in 802.11 wireless networks fall within a region that includes the solutions given by our adaptive power control algorithms. In this region, there is the classical trade-off of throughput versus fairness as we adjust the degree of HN. However, when the degree of HN becomes too high, we enter an *undesirable* "non-trade-off" operating region in which not both fairness and throughput deteriorate simultaneously, even with the reduced EN. The min-transmit-power approach yields solutions belonging to this undesirable operating region.

In summary, there is often a fundamental trade-off between the scalability of network capacity (which is related to EN) and fairness (which is related to HN) in wireless networks. Our experimental results indicate that our algorithms can achieve a good balance between the two in 802.11 networks. In particular, our simulation results show that DAPC and PUSPC on average can improve the network capacity of non-power-controlled 802.11 by more than two times while preserving fairness in the network when the HN-free requirement is imposed. Moreover, PUSPC can simultaneously achieve better fairness and network capacity in the network when compared with the minimum-transmit-power approach.

REFERENCES

[1] W. H. Ho, S. C. Liew, "Achieving Scalable Capacity in Wireless Networks with Adaptive Power Control," the Fifth International IEEE Workshop on Wireless Local Networks (WLN), Nov 2005.

[2] V. Kawadia and P. R. Kumar, "Principles and Protocols for Power Control in Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, pp. 76-88, vol.23, no. 5, Jan 2005.

[3] S. Agarwal, S. V. Krishnamurthy, R. H. Katz, S. K. Dao, "Distributed Power Control in Ad-hoc Wireless Networks," Personal Indoor Mobile Radio Conference, Oct 2001

[4] J. P. Monks, V. Bharghavan, and Wen-mei Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," IEEE INFOCOM 2001.

[5] L. Jiang and S. C. Liew, "Removing Hidden Nodes in IEEE 802.11 Wireless Networks," IEEE Vehicular Technology Conference, Sep 2005.

[6] P. C. Ng, S. C. Liew, "Offered Load Control in IEEE802.11 Multi-hop Ad-hoc Networks," the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Oct 2004.

[7] E. Jung, N. Vaidya, "A Power Control MAC Protocol for Ad Hoc networks," Mobicom 2002.

[8] "The Network Simulator – ns2", http://www.isi.edu/nsnam/ns.

[9] P. C. Ng, S. C. Liew, L. Jiang, "Achieving Scalable Performance in Large-Scale IEEE 802.11 Wireless Networks," IEEE Wireless and Communications and Networking Conference (WCNC'05), Mar 2005.

[10] R. Jain, D. M. Chiu and W. Hawe, "A Quantitative Measure of Fairness and Discrimination For Resource Allocation in Shared Computer Systems," Technical Report TR-301, DEC Research Report, Sep 1984.

[11] A. Sheth, R. Han, "Adaptive Power Control and Selective Radio Activation for Low-Power Infrastructure-Mode 802.11 LANs," International Conference on Distributed Computing Systems Workshops (ICDCSW'03), May 2003.