# Achieving Scalable Capacity in Wireless Networks with Adaptive Power Control *

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

**Abstract** — *The seminar work of Gupta and Kumar [1] showed that multi-hop wireless networks with capacity scalable with the number of nodes, n, are achievable in theory. The transport capacity scales as* $\Theta(\sqrt{n})$*, while the capacity scales as* $\Theta(n)$*. A subsequent study [2], on the other hand, showed that the capacity of IEEE 802.11 networks does not scale with n due to its carrier-sensing mechanism. This prior work, however, has not considered the use of power control. The main contributions of this paper are three-folds: 1) we provide an analytical framework for deriving the design requirements of adaptive power control strategies; 2) we demonstrate that 802.11 networks are scalable with power control; 3) however, an enhanced MAC protocol called Selective Disregard of NAVs (SDN) can achieve substantially higher capacity with an adaptive power control scheme; in particular, adaptive power control allows SDN to achieve capacity within 75% of the theoretical optimal capacity of infrastructure-mode wireless networks. A reason why adaptive power control works well is that it takes into consideration the fundamental mutual-interference relationships between links in the vicinity of each other, and adjust their relative transmit powers to reduce these interferences to a large extent that is possible theoretically.*

## 1. INTRODUCTION

This paper considers the control of transmit power in 802.11 and 802.11-like wireless networks to increase network capacity. In particular, we investigate whether the adjustment of the transmit powers of the nodes in a wireless networks can allow such networks to achieve scalable capacity. This is in contrast to other work [3] [4] in which power control is used to preserve energy and prolong battery life. Having said that, a desirable offshoot of our approach is that battery life may also be prolonged because maximal capacity is typically achieved when low transmit powers are used so that high spectrum re-use can be achieved.

Whether network capacity is scalable as node density increases is directly related to how many links in the network can transmit simultaneously without interfering with each other. There are two types of constraints against the success of simultaneous transmissions by links [2]: namely, the physical-collision constraints, and the protocol-collision-prevention constraints. Figure 1 shows the classification of transmission constraints.

The physical-collision constraints are due to the receiver's inability to decode its signal when the powers received from other interfering sources are large. So, if two links that do not fulfill the physical-collision constraint transmit simultaneously, one or both of the transmissions will fail.

Physical-collision constraints can be further divided into two classes: protocol-independent and protocol-specific physical-collision constraints. The protocol-independent constraints are independent of the multi-access protocol used. They only require the Signal-to-Interference Ratio (SIR) at the receiver of a link to be sufficiently large to be collision-free.

Protocol-specific physical-collision constraints are additional collision constraints due to the protocol used. For example, in the 802.11 MAC protocol, after the reception of a DATA frame by the receiver, the receiver will immediately return an ACK to the transmitter. This ACK must be properly received at the transmitter. In addition, this ACK should not cause collisions at other simultaneously transmitting links. Thus, the SIR of the ACK at the transmitter must be sufficiently large, and the interference induced by the ACK at other links sufficiently small. That is, there should be no DATA-ACK and ACK-ACK collisions either if simultaneous transmissions are to be successful. It has been shown that based on physical-collision constraints alone, the network capacity is scalable with *n,* where *n* is the number of nodes [1] [2] in a fixed area, with order $\Theta(n)$.

The protocol-collision-prevention constraints are due to the operation of the multi-access protocol to prevent collision-causing simultaneous transmissions described in the aforementioned physical-collision constraints. In 802.11 networks, this is achieved through the carrier-sensing mechanism. However, in preventing illegitimate simultaneous transmissions, the protocol may become overly aggressive and may disallow some simultaneous transmissions that are non-collision-causing. Simultaneous transmissions disallowed by a protocol can be encapsulated in a set of protocol-collision-prevention constraints, which can be further divided into transmitter-side and receiver-side carrier-sensing constraints, as shown in Figure 1. They will be elaborated in Section 2.2. The 802.11 protocol-collision-prevention constraints, in particular, cause the 802.11 network capacity to be non-scalable [2]. Take 802.11 infrastructure networks for example, beyond a certain point, increasing the number of access points (AP) in a fixed area does not increase the overall network capacity at all.

A variant of 802.11 for achieving scalable capacity, *Selective Disregard of NAVs (SDN)*, was introduced in [2]. The main idea of SDN is to eliminate the extraneous 802.11 protocol collision-prevention constraints so that the capacity is limited only by the more fundamental physical-collision constraints. That is, the SDN's protocol-collision-prevention constraints overlap with the physical-collision constraints. It has been shown that without power control, the original

* This work was sponsored by the Areas of Excellence scheme established under the University Grant Committee of the Hong Kong Special Administrative Region, China (Project Number AoE/E-01/99).

802.11 networks are non-scalable, but SDN networks are [2].

The previous work, however, has not considered the use of power control. This paper considers the implication of power control for capacity scalability in 802.11 and SDN wireless networks. Specifically, we investigate 1) a simple power control scheme in which the transmit powers of all transmitters are uniformly scaled; and 2) an adaptive power control scheme in which different transmitters may use different powers to further boost capacity.

The remainder of the paper is organized as follows. Section 2 introduces the transmission constraints of SDN and 802.11 with power control. Section 3 proposes a graph model for the systematic study of the constraints. The detailed operation of SDN is given in Section 4. Section 5 discusses the scalability of capacity in SDN and 802.11 with uniformly-scaled power control. The adaptive power control scheme is introduced in Section 6. Finally, Numerical results from simulations are presented in Section 7, before the conclusion in Section 8.

## 2. SIMULTANEOUS-TRANSMISSION CONSTRAINTS WITH POWER CONTROL

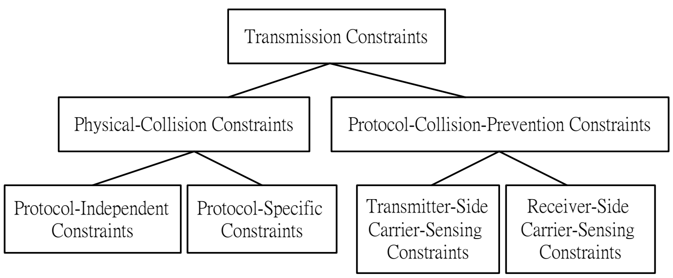


Figure 1. Classification of transmission constraints in 802.11-like networks.

### 2.1 Transmission Constraints of SDN

SDN removes the extraneous 802.11 protocol constraints (i.e., constraints against simultaneous transmissions that are non-collision-causing) so that the capacity is largely limited only by the physical-collision constraints. Details of SDN will be presented in Section 4. From the viewpoint of transmission-constraint, the protocol-collision-prevention constraints in SDN basically track the physical-collision constraints.

We will model the physical-collision constraints using the pair-wise interference model [1]. We assume the power-transfer relationship from node $a$ to node $b$ is $P(a,b) = k \cdot P_a / r^{\alpha}$, where $P(a, b)$ is the power received by node $b$ from the transmission by 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 $i$. With no power control, all transmit powers are the same, so that for all $i$, $P_{Ti} = P_T$, a constant. For brevity, we will also use $T_i$ and $R_i$ to denote their positions. So, $|a - b|$ denotes the physical distance between nodes $a$ and $b$.

**A. Protocol-Independent Physical-Collision Constraints**

Consider two links, 1 and 2. The power received by $R_1$ from $T_1$ must be sufficiently larger than the power received by $R_1$ from $T_2$ in order that the signal from $T_1$ can be successfully decoded. Let the Signal-to-Interference requirement be $K$ (e.g., 10 dB). Then, collision occurs when $P(T_1, R_1) < K\, P(T_2, R_1)$. Plugging in the power-transfer relationship $P(a, b) = kP_a / |a\text{-}b|^{\alpha}$, we have

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

**B. Protocol-Specific Physical-Collision Constraints**

In SDN, each data transfer on link $i$ consists of a DATA frame in the forward direction and an ACK frame in the reverse link $i'$ in the other direction. This induces physical-collision constraints due to the presence of ACK on the reverse link $i'$.

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

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

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

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.

### 2.2 Transmission Constraints of 802.11

Both 802.11 and SDN share the same physical-collision constraints as described in Subsections 2.1A and 2.1B. To prevent such collisions, each operates a multi-access protocol to prevent these collision-causing simultaneous transmissions.

The simultaneous transmissions disallowed by the protocols can be expressed in terms of a set of collision-prevention constraints. The inequalities that encapsulate SDN's collision-prevention constraints are a subset of the inequalities that encapsulate collision constraints. So, there is no need to discuss additional constraint inequalities due to the collision-prevention measure in SDN. The reader is referred to Section 4 on how this is achieved.

The standard 802.11, on the other hands, has collision-prevention constraints that are distinct from the physical-collision constraints. In particular, 802.11 collision-prevention constraints may fail to prevent some collision-causing transmissions while preventing some non-collision-causing transmissions. This section is devoted to the discussion of 802.11 collision-prevention constraints.

Consider links 1 and 2 again, and suppose that link 1 is transmitting. Then link 2 cannot transmit if any of the transmitter-side and receiver-side carrier-sensing constraints in Subsections A and B holds.

**A. Transmitter-Side Carrier-Sensing Constraints**

The transmitter-side carrier-sensing constraints are induced by the use of RTS/CTS and PHY preamble access mode in 802.11, which are originally designed to prevent collisions. Transmitter $T_2$ cannot transmit if it senses the ongoing transmission on link 1. This will be the case if any of the following holds:

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

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

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

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 $a$.

**B. Receiver-Side Carrier-Sensing Constraints**

Now, if none of (5) − (7) holds, $T_2$ may go ahead to transmit DATA (or RTS). However, if any of (8) − (10) is satisfied, $R_2$ will not reply an ACK (or CTS) to $T_2$, causing $T_2$ to interpret that as a collision, and that will start off the 802.11 MAC backoff algorithm and retransmission procedure.

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

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

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

The rationale for $R_2$ not returning an ACK (or CTS) to $T_2$ is that $R_2$ can sense the ongoing transmission on link 1, and the ACK may interfere with the ongoing transmission on link 1.

There is also another set of inequalities similar to (5) − (10) for the case that link 1 tries to transmit while link 2 is already transmitting. The case of no power control is a special case in which $VCSRange(P_a)=VCSRange$ and $PCSRange(P_a)$ $=PCSRange$, where $VCSRange$ and $PCSRange$ are constants.

## 3. INVESTIGATION OF TRANSMISSION CONSTRAINTS USING GRAPH MODEL

### 3.1 Link-Interference Graph from Physical-Collision Constraints

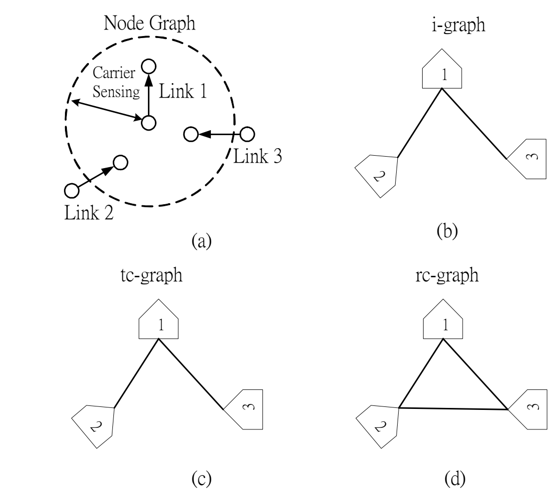


Figure 2. Mapping of a network topology a) to b) i-graph, c) tc-graph and d) rc-graph.

A *Link-Interference Graph (i-graph)* can be used to represent the physical-collision constraints graphically. In an i-graph, an arrow-shape vertex represents a wireless link with the arrowhead pointing toward the receiver. There is a directional interference edge (i-edge) from vertex 2 to vertex 1 if any of the constraints (1) − (4) is satisfied. Figures 2a and 2b show the mapping from a network topology to an i-graph.

### 3.2. Protocol-Collision-Prevention Graphs of 802.11

We propose the use of two protocol-collision- prevention graphs to model the constraints against simultaneous transmissions introduced by the protocol. They are respectively the tc-graph and rc-graph. In the tc-graph, there is a directional tc-edge from vertex 1 to vertex 2 if any of the inequalities (5) − (7) is true. A tc-edge from vertex 1 to vertex 2 means that a current transmission on link 1 can and will warn link 2 not to transmit before its completion.

In the rc-graph, there is a directional rc-edge from vertex 1 to vertex 2 if any of the inequalities (8) − (10) is satisfied. Link 2 can still transmit if there is a rc-edge from vertex 1 to vertex 2 but no corresponding tc-edge; however, the fact that the receiver will ignore the DATA frame and not return an ACK means the transmission by link 2 will fail.

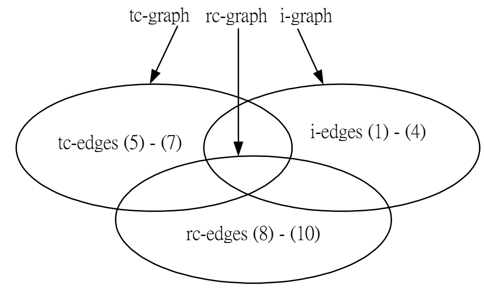


Figure 3. Relationships among i-edges, tc-edges and rc-edges in i-graph, tc-graph and rc-graph; and the constraints associated with the edges.

Figures 2c and 2d show an example of a tc-graph and rc-graph respectively. Consider links 2 and 3, an edge is drawn in the rc-graph but not in the tc-graph and i-graph. Since the receivers of links 2 and 3 are within the carrier-sensing range of each other's, there is a rc-edge between links 2 and 3. Figure 3 shows the relationships among different types of edges and the inequalities that define them.

## 4. SELECTIVE DISREGARD OF NAVS (SDN)

This section elaborates on the SDN protocol. As has already been mentioned, SDN is scalable without power control. And as will be discussed in Section 7, its performance is even better with power control.

There are three parts to SDN, described as follows. We shall use the term SDN loosely to refer to the collection of the three parts, as well as SDN.II specifically. It is SDN.I and SDN.II that remove the extraneous carrier-sensing constraints in standard 802.11 that are limiting the scalability of the network. SDN.III is an accessory to SDN.II so that a node has enough information to decide whether to ignore an ongoing transmission during its carrier-sensing operation.

**SDN.I – Turning off Physical Carrier Sensing and using Receiver Restart Mode:** Physical carrier sensing is deactivated. In addition, Receiver Restart Mode is turned on. In some commercial 802.11 chips, there is a so-called *"Restart Mode" (RS)* in the receiver design. If the receiver is in the midst of receiving a signal, another signal with sufficiently larger power relative to the first signal arrives, the receiver will switch to receive the new signal. With SDN.I, constraints (7) and (10) are removed. In particular, the associated tc- and rc-edges are eliminated.

**SDN.II – Selective Disregard of NAVs (SDN):** Virtual carrier sensing of 802.11 is modified so that RTS/CTS not just contain the address of the receiver, but also that of the source. Each node not just monitors whether the wireless channel is busy, but also who is transmitting to whom. A node will decide to go ahead to transmit even if the medium is busy, provided the currently transmitting links are non-interfering. For example, in the original 802.11, according to (5) and (6), $T_2$ will not transmit if it already hears the RTS or CTS on link 1. However, with SDN, $T_2$ will transmit if it finds that (1) – (4), as well as (1) – (4) with the variables $T_1$ and $R_1$ interchanged with $T_2$ and $R_2$, do not hold – i.e., links 1 and 2 do not mutually interfere physically.

Similarly, according to (8) and (9), in the original 802.11, $R_2$ will not reply a CTS to the RTS of $T_2$ if $R_2$ already hears the RTS or CTS on link 1. However, with SDN, $R_2$ will return a CTS if it finds that (1) – (4), as well as (1) – (4) with $T_1$ and $R_1$ interchanged with $T_2$ and $R_2$, do not hold.

Let us denote the set of i-edges by $I$, the set of tc-edges by $TC$, and the set of rc-edges by $RC$. Together with SDN.I, SDN.II removes the effects of the edges in ($|TC \cup RC| - |(TC \cup RC) \cap I|$), which encapsulates the number of extraneous collision-prevention constraints induced by the protocol that are preventing legitimate pair-wise simultaneous transmissions. For details of the whole algorithm, the reader is referred to [2].

**SDN.III – Constructing i-graph using Power Exchange Algorithm (PE):** For the operation of SDN.II, a node needs to know its interference relationships with its neighbor as described by (1) – (4). That is, it needs to know its i-edges with its neighbors. To discover such relationships, each node exchanges power-transfer information with its neighbors through a power-exchange algorithm. The reader is referred to [2] for details of the power-exchange algorithm for gathering information required for SDN operation. The modification of the PE algorithm for gathering information required for adaptive power control will be explained in Section 6.

## 5. SCALABILITY OF NETWORK CAPACITY: ANALYTICAL DISCUSSION

We now present a rough intuitive discussion of the scalability of network capacity with and without power control. Consider an infinitely large infrastructure-mode wireless network with multiple access points (AP) laid out in a grid topology. Each grid is associated with an AP, which serves as the base station for the clients located within the grid. Clients are randomly placed within the grid. We consider the total network capacity when we increase the number of APs by reducing the grid size, while maintaining the same number of clients per AP.

The increase of node density as such is analogous to decreasing the scale on which a map is drawn (see Figure 4). The location ($x$, $y$) is translated to ($ax$, $ay$), where $a < 1$. If the distance between two nodes before scaling is $d$, the distance after the transformation is $ad$.

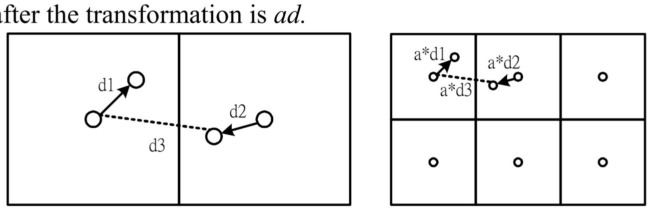

Figure 4. Change in node density as the network scales.

For SDN without power control, constraints (1) – (4), with $P_{T1} = P_{R1} = P_{T2} = P_{R2}$ are the only constraints. When these powers are the same, (1) – (4) are invariant to this scaling, as $a$ on both sides of the inequalities will cancel out. Thus, the expected *network capacity per unit area* will scale with the number of APs. This is because the expected capacity of a smaller grid is the same as the expected capacity in the original larger grid before transformation.

On the other hand, 802.11 networks without power control are limited by constraints (1) – (10) with $P_{T1} = P_{R1} = P_{T2} = P_{R2}$. Since the carrier-sensing ranges are constant as we scale, they will cover more neighbor grids when the node density increases (Figure 5). Thus, the network capacity per unit area will reach a limit eventually as the number of APs increases. That is, *more APs do not bring about higher capacity!* This is the behavior we expect from today's 802.11 products since they do not support transmit power control.

Intuitively, if power control is introduced so that *VCSRange* and *PCSRange* are also scaled according to the size of grids, some of the extraneous inter-grid carrier sensing can be prevented and we should observe network capacity that scales with node density.

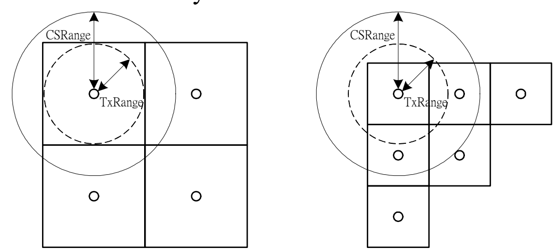

Figure 5. Effect of *CSRange* in 802.11 as the network scales.

We define two types of power control here. The first is *Uniformly-Scaled Power Control (USPC)*, in which all nodes use the same transmit power adjusted in accordance with node density. The second is *Adaptive Power Control (APC)*, in which different nodes may use different transmit powers that are adapted to the locality of their surrounding links.

With USPC, $P_i$ will still be constant for all nodes $i$. Constraints (1) – (4) with $P_{T1} = P_{R1} = P_{T2} = P_{R2}$ remain the same. The resulting SDN network capacity should be similar to that without power control. The implication is that we need APC to boost the capacity of SDN.

For 802.11 with USPC, we need to consider constraints (1) – (10). We would expect the capacity in 802.11 to increase with USPC as per our discussion of Figure 5. However, we would still expect 802.11 with USPC to have smaller network capacity than SDN without power control. To see that, note that with USPC, constraints (1) – (4) remain the same, with $P_{T1} = P_{R1} = P_{T2} = P_{R2}$. These are the same set of constraints characterizing SDN. However, 802.11 with USPC is additionally constrained by (5) – (10). Therefore, we would expect SDN without power control to perform better. The reader is referred to Section 7 for the experimental results.

## 6. ADAPTIVE POWER CONTROL (APC)

There are two shortcomings to USPC: 1) With respect to SDN, it does not bring about any capacity advantage. 2) It may be suboptimal in scenarios in which links are non-uniformly distributed, as well as in scenarios in which power propagation is not a regular function of distance. Specifically, in USPC the transmit powers of all nodes are kept the same, and a link's transmit power is not adapted to what the link sees from other links in its neighborhood. The goal of APC is to overcome these limitations.

### 6.1 Adaptive Power Control for SDN

We now present the details of our APC algorithm for SDN. The APC algorithm for 802.11 will be presented in the next subsection by modifying the APC for SDN.

The execution of APC algorithm consists of successive iterations. We assume that initially the transmit powers of nodes are high. Each iteration chooses a particular link and attempts to reduce the number of i-, tc-, and rc-edges emanating out of that link by reducing the transmit powers of its transmitter and receiver. Whereas in SDN, the sets of tc- and rc-edges are inside the set of i-edges.

There are two issues: (i) in each iteration, how to adjust the transmit power of the chosen link; and (ii) which link should be chosen for each iteration. In addition, to address both (i) and (ii), it is necessary to find out the transmit powers used by the nodes, $\{P_i\}$ and the gain matrix among the nodes $[G(i, j)]$, where $G(i, j) = P(i, j) / P_i$ is the power-gain function from node $i$ to node $j$.

This subsection is divided into four parts. In Part A, we discuss how to determine the maximum amount of the transmit powers of a link can be adjusted down by – issue (i) above. Part B considers the order in which nodes adjust their transmit power and its impact on the network capacity – issue (ii) above. In Part C, we introduce an algorithm for nodes to exchange power-transfer information so that they can obtain the $\{P_i\}$ and $G(i, j)$ in their neighborhood for the calculations in Part A and Part B. Part D presents simulation results of the different strategies proposed in Part B.

#### A. Per-iteration Power Adjustment

We now consider the power adjustment of a chosen link. To reduce the i-edges from a link to other links, we may reduce the transmit powers of the transmitter and receiver of that link. The larger the values of such transmit power reductions, the more likely that more i-edges can be eliminated. However, there is a bound on the transmit powers that can be reduced, as explained below.

In the *Per-iteration Power Adjustment* of APC, we assume the transmit powers of links other than the chosen link remain unchanged, and we only reduce the transmit powers of nodes. When adjusting the transmit power of the chosen link, we must make sure that 1) the connectivity between its transmitter and receiver can be maintained; and 2) the power reduction does not create new i-edges from other links to chosen link. Note that reducing power as such will not create new i-edges from the chosen link to other links.

1. **Ensuring the reduced power satisfy the minimum decodable threshold:** Suppose $T_1$ is the sender and $R_1$ is the receiver of link 1. Suppose that $G(T_1, R_1)$ and $G(R_1, T_1)$ are known (this can be achieved with the Power Exchange Algorithm described in Part C). Then, to guarantee connectivity from $T_1$ to $R_1$, the minimum transmit power of $T_1$ must be bounded below by

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

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 $j$. Similarly, the minimum transmit power of $R_1$ must be bounded below by

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

2. **Ensuring the *VCSRange* is enough to cover interfering nodes:** This requirement is to ensure that virtual carrier sensing in SDN continues to work well. Let $M_{T1}$ denote the set of nodes whose transmissions can interfere with the reception at $T_1$. So, before $T_1$ transmits, it has to be able to warn the nodes in $M_{T1}$ not to transmit via virtual carrier sensing. Otherwise, the ACK from $R_1$ to $T_1$ might be corrupted by transmissions by the nodes in $M_{T1}$. This can be achieved in two ways. Either the RTS of $T_1$ or the CTS of $R_1$ must reach the nodes in $M_{T1}$. Thus, we have

$$P_{adjusted}(T_1) \;\geq\; Rx_{th}^{VCS} / G(T_1, m) \;\; \forall m \in M_{T1} \quad \textbf{(13)}$$

OR

$$P_{adjusted}(R_1) \;\geq\; Rx_{th}^{VCS} / G(R_1, m) \;\; \forall m \in M_{T1} \quad \textbf{(14)}$$

where $Rx_{th}^{VCS}$ is the receiver sensitivity threshold for RTS/CTS which is generally smaller than $Rx_{th}$ so that *VCSRange* is larger than *TxRange*. For example, if RTS/CTS are transmitted at $1/r_{VCS}$ the rate of DATA, as an approximation, we may set $Rx_{th}^{VCS} = Rx_{th} / r_{VCS}$.
Note that in the above: (i) We have assumed the same transmit power is used to carry DATA/ACK and RTS/CTS. (ii) (13) and (14) are an OR relationship. When adjusting the transmit powers of $T_1$ and $R_1$, as long as one of them is satisfied, the condition is fulfilled.

3. **Ensuring the reduced power is stronger than neighbors' pair-wise interferences:** To ensure that *no new i-edges will be established when $T_1$ and $R_1$ reduce their transmit powers*, let $N_{R1}$ and $N_{T1}$ be respectively the set of transmitting and receiving nodes in neighboring links that are not attacking $R_1$, and $T_1$ originally, but which potentially may do so if the power adjustment is too aggressive. We require

$$P_{adjusted}(T_1) \geq KP_nG(n, R_1) / \mathrm{G}(T_1, R_1) \;\; \forall n \in N_{R1} \quad \textbf{(15)}$$

$$P_{adjusted}(R_1) \geq KP_nG(n, T_1) / \mathrm{G}(T_1, R_1) \;\; \forall n \in N_{T1} \quad \textbf{(16)}$$

Note that $N_{R1}$ and $N_{T1}$ do not need to cover all nodes in the network. In particular, they need to cover only nodes that can potentially interfere with $R_1$ and $T_1$ respectively. For this, only node $n$ that satisfies the following needs 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 $P(n, R_1) \;\geq\; Rx_{th} / K$

Steps 1, 2, and 3 are combined as follows. First, we set $P_{adjusted}(T_1)$ to the maximum of (11) and (15). Then, we set $P_{adjusted}(R_1)$ to the maximum of (12) and (16). Then, we see if either $P_{adjusted}(T_1)$ satisfies (13) or $P_{adjusted}(R_1)$ satisfies (14). If yes, we are done. If not, we adjust one of $P_{adjusted}(T_1)$ or $P_{adjusted}(R_1)$ upward until either (13) or (14) is fulfilled.

In general, the computation time for each per-iteration power adjustment is $O(n)$, where $n$ is the number of nodes, thanks to steps 2 and 3.

**B. Power Control Scheduling Strategies**

We would like to study the importance of the order of links for power control, based on a link-by-link power adjustment nature. We consider the strategies for choosing a link for power adjustment in each iteration – referred to as *Power Control Scheduling Strategies*. Specifically, three strategies are considered. Their performance results are presented in Part D.

In this paper, we assume there is a central node that knows the power-transfer relationships among links and decides which link to control its power in each iteration. In real practice, we will implement APC in a distributed manner, in which no central node is needed. In distributed algorithms, every node only needs to monitor the local conditions surrounding them and multiple nodes may adjust their powers simultaneously. It turns out the distributed versions of the centralized algorithms discussed here can be easily devised. Due to space limitation, detailed discussions of the distributed algorithms are relegated to a separate paper. Generally speaking, good centralized algorithms also yield good distributed algorithm. In addition, it is also essential to understand centralized algorithms as benchmarks even though our ultimate goal is distributed versions of them.

Let us denote the number of attacking i-edges of link $l$ by $i_a(l)$ – i.e., number of i-edges from $l$ to other links; and the number of defending i-edges of link $l$ by $i_d(l)$ – i.e., number of i-edges from other links to $l$.

**Strategy 1 – Choose the link with the largest $i_a$:** The intuition of this strategy is as follows. The link with the largest number of attacking i-edges is the link that seriously interferes with neighboring links. By reducing its power first, we can increase the chance that more i-edges can be reduced in the iteration. Note, however, that the power adjustment steps in the previous subsection is a defensive one in that it ensures that no new i-edges to the chosen link is created, rather than that some i-edges from it to others are eliminated. Thus, this strategy does not guarantee to reduce the largest number of i-edges in each iteration.

In case of a tie in which multiple links have the same $i_a$, we will pick the one with the smallest $i_d$. If there are multiple links with the same $i_a$ and $i_d$, one of the links will be chosen in random. The reader is referred to the description of Strategy 2 for the motivation for considering a link with the smallest $i_d$.

As a side note, instead of choosing the links with the largest $i_a$, in the whole network, the distributed version of the algorithm have many links choosing the largest $i_a$, in their neighborhoods for simultaneous power adjustment. Similar comments apply to other strategies discussed later.

The overall centralized APC algorithm of Strategy 1 is shown in Figure 6. Once a link is selected for power adjustment, it will not be selected again. A "round" consists of the considerations of all links. We choose a link only once in each round because otherwise it is possible for the chosen link to be chosen again in the next iteration because it still has the largest $i_a$. This will result in an infinite loop. We may run the algorithms for several rounds to continue to reduce the i-edges. In our simulation experiment, however, we have found that typically after one round, only a few additional i-edges can be eliminated in future rounds.

```
Pseudocode of Strategy 1:
//LinkSet is the set for link waiting for power control
LinkSet = all links;
While (LinkSet != NULL){
        L = arg max_l in LinkSet (i_a(l));
        If |L|>1, L = arg min_l in L (i_d(l)) else m = link in L;
        If |L|>1, m = a random link in L;
        Perform per-iteration power adjustment on m;
    Remove m from LinkSet
}
```

Figure 6. Pseudocode of Strategy 1.

Since the computation time of per-iteration power adjustment is $O(n)$, and Strategy 1 loops for $l$ iterations, where $l = cn$ is the number of links for some constant $c$. Thus, the computation time for one round of Strategy 1 is $O(n^2)$.

**Strategy 2 – Choose the link with the smallest $i_d$:** The intuition of this strategy is as follows. With respect to step 2 of the per-iteration power adjustment algorithm in Part A, having fewer defending i-edges to consider may allow us to lower the power by a larger amount. Hence, this may increase the likelihood of an attacking i-edge being eliminated. That is, whereas Strategy 1 maximizes the number of candidate i-edges for elimination, Strategy 2 maximizes that the chance that a candidate i-edge can be eliminated. Neither strategy, however, guarantees that an i-edge can in fact be eliminated in each iteration.

In case of a tie in which multiple links have the same $i_d$ we will pick the one with the largest $i_a$, If there are multiple links with the same $i_a$ and $i_d$, one of the links will be chosen in random.

```
Pseudocode of Strategy 2:
//LinkSet is the set for link waiting for power control
LinkSet = all links;
While (LinkSet != NULL){
        L = arg min_l in LinkSet (i_d(l));
        If |L|>1, L = arg max_l in L (i_a(l)) else m = link in L;
        If |L|>1, m = a random link in L;
        Perform per-iteration power adjustment on m;
    Remove m from LinkSet
}
```

Figure 7. Pseudocode of Strategy 2.

The overall APC algorithm for Strategy 2 is shown in Figure 7. Similar to Strategy 1, once a link is picked for power control, it will not be picked again in each round to avoid infinite looping of the algorithm. As with Strategy 1, the computation time of Strategy 2 is also $O(n^2)$ in one round due to the computation of $l$ per-iteration power adjustment.

**Strategy 3 – Choose the link that maximizes the number of i-edges than can be eliminated in this iteration:** Strategies 1 and 2 do not guarantee to reduce the largest number of i-edges in every iteration. Strategy 3 is a greedy algorithm that tries to optimize every step.

We need to define the *reducibility* of a link. An i-edge between two links is said to be reducible if it ceases to satisfy any of the constraints (1) – (4) after per-iteration power adjustment in Part A is performed. Let $i_r(l)$ denote the number of i-edges that is reducible if link $l$ is chosen for power adjustment. For this strategy, we first calculate the adjusted transmit power of every link and their respective $i_r$. Then we pick the one with the largest $i_r$ for power reduction.

If there is more than one link with the maximum $i_r$, we will pick the one with the smallest $i_d$ to break the tie. The overall APC algorithm of Strategy 3 is shown in Figure 8. Unlike Strategies 1 and 2, Strategy 3 allows links that have been chosen to be chosen again – there will be no infinite loop like that described previously. So, the number of iteration could be greater than the number of links $l$ in each round.

```
Pseudocode of Strategy 3:
//LinkSet is the set for link waiting for power control
LinkSet = all links;
While (i_r_max != 0){
        for (l=1; l<|LinkSet|; l++){
        // i_r(l) is the number of reducible i-edges of link l
            find i_r(l);
        }
        // for all l belongs to LinkSet
        L = arg max_l in LinkSet (i_r(l));
        If |L|>1, L = arg min_l in L (i_d(l)) else m = link in L;
        If |L|>1, m = a random link in L;
        // i_r_max stores the max i_r among links
        i_r_max= i_r(L);
        Perform per-iteration power adjustment on L;
}
```

Figure 8. Pseudocode of Strategy 3.

In Strategy 3, the algorithm can loop until no more i-edge is reducible. Since there are at most ${}_lC_2 = l(l\text{-}1)/2$ potential i-edges in a network with $l$ links, in the worst-case, there are $O(n^2)$ iterations. The process of finding the number of reducible i-edges and per-iteration power adjustment for links both consume $O(n)$ computation time. Thus, the total computation of Strategy 3 is $O(n^3)$.

### C. Power Exchange Algorithm

In [2], a Power Exchange Algorithm (PE) has been proposed for establishing the i-graph of a network. Our per-iteration power adjustment procedure in Part 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 no new i-edges are created after power adjustment. The power control scheduling strategies proposed in Part B also require enough information to elect the link for power control. We extend the PE in [2] for our purpose here.

The PE algorithm is a local algorithm in that each node finds out the i-edges and potential i-edges in its neighborhood. If we assume the presence of a central node, such information can be gathered for the algorithms in Parts A and B.

Receiver Restart Mode in the receiver design is assumed. 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: (1) Active links ($a$, $b$) or ($b$, $a$), where $b$ is any other node which forms an active link with $a$; (2) 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) (3) "Power set", as described below. The identity of the sender of a PE packet is implicit in the MAC address of the PE packet.

1. Each node $a$ monitors the power it receives from other nodes and keep this information in a power set $PS_a = \{[b, P(b, a)], [c, P(c, a)], \ldots\}$ where $P(i, j)$ denotes the power received at node $j$ from node $i$. For this purpose, the powers from the PE packets from nodes $b$, $c$, … can be measured by node $a$.
2. Each node $a$ periodically broadcasts a PE packet at a rate lower than the data rate.
3. Node $a$ gathers information from the PE packets received from its neighbors.

The following condition is sufficient to ensure that the necessary information for Parts A and B can be gathered by the above PE algorithm:

$$Rx_{th}^{PE} < Rx_{th} \;/\; K \qquad \textbf{(17)}$$

where $Rx_{th}^{PE}$ is the receiver sensitivity threshold for PE packets. Note that according to (17), if $d_{max} = TxRange$, where $d_{max}$ is the maximum link distance, then PE packets must be transmitted at a lower rate than DATA packets in order that $PERange > TxRange$.

***Proof of Correct Operation of PE for Part A***

We will first prove that enough information can be gathered for step 3 of the per-iteration power adjustment in Part A through PE, and this proof implies the correct operation of steps 1 and 2.

For the execution of step 3, with respect to link 1, we need to know $G(n, T_1)\ \forall n \in N_{T1}$, $G(n, R_1)\ \forall n \in N_{R1}$, and $G(T_1, R_1)$. Suppose we count on $T_1$ and $R_1$ to gather these data based on the PE packets.

***Determination of G(T₁, R₁)***

$T_1$ can measure the power of the PE packets from $R_1$ and by examining the content of the PE packets to find out $P_{R1}$, $T_1$ can then determine $P(R_1, T_1) / P_{R1} = G(R_1, T_1) = G(T_1, R_1)$ by symmetry.

***Determination of G(n, T₁) and G(n, R₁)***

According to step 3, node $n$ satisfies either

$$P(n, R_1) > Rx_{th} / K \quad \forall n \in N_{R1} \text{ or } P(n, T_1) > Rx_{th} / K \quad \forall n \in N_{T1}$$

Substituting (17) into the above, we have

$$P(n, R_1) > Rx_{th}^{PE} \quad \forall n \in N_{R1} \text{ or } P(n, T_1) > Rx_{th}^{PE} \quad \forall n \in N_{T1}$$

So, the above says that the PE packets of nodes that may potentially interfere with $T_1$ and $R_1$ can reach $T_1$ and $R_1$. Consider $T_1$. $T_1$ can measure the power of the PE packet from node $n$ in $N_{T1}$, and by examining its content, $T_1$ can derive $G(n, T_1)$. Similarly, $R_1$ can find out $G(n, R_1)\ \forall n \in N_{R1}$.

Let us now look at steps 1 and 2. In step 1, we need $G(T_1, R_1)$ and $G(R_1, T_1)$ for calculating the minimum necessary transmit power of $T_1$ and $R_1$. But it has been shown that $T_1$ can determine $G(T_1, R_1) = G(R_1, T_1)$.

In step 2, we need $G(T_1, m)$ and $G(R_1, m)\ \forall m \in M_{T1}$. It has been shown in the proof of step 3 that both $G(n, T_1)$ and $G(n, R_1)$ can be determined by $T_1$ and $R_1$ respectively from the PE packets for all nodes $n$ that satisfies:

$$P(n, R_1) > Rx_{th} / K \quad \forall n \in N_{R1} \text{ or } P(n, T_1) > Rx_{th} / K \quad \forall n \in N_{T1}$$

In step 2, $M_{T1}$ denotes the set of nodes whose transmissions

can interfere with reception at node $T_1$. So, it is trivial that
$P(m,T_1) > Rx_{th} / K \quad \forall m \in M_{T1}$

Therefore, $G(T_1, m) = G(m, T_1)$ can be determined by $T_1$.

If $T_1$ is closer to node $m$ than $R_1$, we can focus on satisfying (13) and we are done, since only (13) or (14) needs to be satisfied. If $R_1$ is closer to node $m$ than $T_1$, then $R_1$ will also be interfered by node $m$, and according to the above, $R_1$ can determine $G(R_1, m) = G(m, R_1)$. So, the algorithm can use both (13) and (14) to see which is easier to satisfy. In any case, we will have the sufficient information from $T_1$ and $R_1$.

***Proof of Correct Operation of PE for Part B***

For Strategy 1 and Strategy 2, we need to know the number of i-edges in each link.

In the proof of correct operation of PE for step 2 in Part A, we have shown that whenever there is a node $m$ whose transmission interferes with the reception at $T_1$, the PE packet from node $m$ can reach $T_1$ provided (17) is satisfied. $P(m, T_1)$ can be found from the power set in the PE packet. Similarly, whenever there is a node $m$ whose transmission interferes with $R_1$, the PE packets from node $m$ can reach $R_1$, so that $P(m, R_1)$ can be found from the power set. Also, it has been proved that $T_1$ and $R_1$ can determine $P(R_1, T_1)$ and $P(T_1, R_1)$ respectively. Hence, the information gathered by $T_1$ and $R_1$ is enough for us to determine all the i-edges to the links $(T_1, R_1)$ and $(R_1, T_1)$. Since we assume in this paper that all nodes will send such information to the central node, it can thus determine all the i-edges in the network.

For Strategy 3, we need to compute the number of reducible i-edges by power adjustment. This requires the power-transfer relationships of the existing i-edges, potential i-edges, and the current transmit powers used. It has already been shown that such power-transfer relationships can be gathered by the PE algorithm. So, if the nodes also send the current transmit powers they use, the central node will have all the information required for Strategy 3.

**D. Comparison of Scheduling Strategies**

We now present the performance results of the three strategies. We established in MATLAB an infrastructure topology with 25 APs placed uniformly in a square-grid manner over a 1x1 $km^2$ domain. Then, 125 client stations are placed randomly in the domain so that each AP on average has five associated clients. In addition to the three strategies, for benchmarking purposes, we also consider the random strategy in which a random link is chosen in each iteration.

Figure 9 shows the number of remaining i-edges versus the number of per-iteration power adjustments performed. The number of iterations for Strategies 1 and 2 are equal to the number of links (i.e., 125), while Strategy 3 can have more iterations until no more i-edge is reducible. It can be seen that the number of remaining i-edges of the random case is always more than that of the three strategies, and Strategy 3 performs the best followed by Strategy 2 and then Strategy 1.

It is interesting to note that for gaps between the remaining i-edges widen and then narrow as the number of iterations increases. Specifically, the remaining i-edges of the different strategies converge to values that differ by about 40 i-edges. At convergence, the different strategies may not yield significantly different network capacities.

The convergence rate is an important factor to consider if the power-transfer relationships between nodes are an attribute that can change dynamically over time (e.g., mobile nodes or dynamically changing environment in a static network). For the case with mobility, we can quantize the time by iterations of the strategy. If the nodes move less frequently, more iterations of the power control strategy can be performed and higher capacity can be achieved

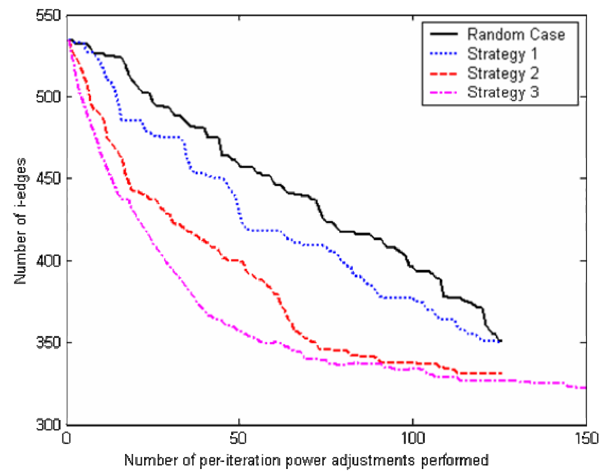


Figure 9. Number of remaining i-edges versus number of iterations.

We emphasize again that in real practice, APC will be implemented in a distributed manner, every link only monitors its local surroundings, and powers used by links that are far apart can be adjusted simultaneously without coordination among them. Each link only coordinates and exchanges information with a subset of nearby links. As a result, the actual number of iterations needed to achieve a reasonably good performance should be smaller.

**6.2 Adaptive Power Control for 802.11**

APC for 802.11 needs to consider the effects of tc-edges and rc-edges because unlike in SDN, there could be many tc-edges and rc-edges without corresponding i-edges. This will be the case, for example, if the initial transmit powers used by the nodes are much higher than that is required for the maximum transmission range between transmitters and receivers of links. Many legitimate simultaneous transmissions of links without i-edges among them will then be prevented by the carrier-sensing mechanism.

If we know the network topology (e.g., the grid topology used in our simulations), we could first apply USPC to reduce the transmit powers to a certain extent before applying APC. This will be the strategy we adopt in our investigation here. In the case of the infrastructure-mode grid topology, we scale the *TxRange* of APs to just cover a grid area initially to ensure links could be established with the minimum power. The initial transmit power of stations is set to be the same as that of the APs. Of course, we could apply APC directly without USPC beforehand. Our simulation experiments, however, indicate that many rounds of the execution of the APC algorithm will then be needed for convergence. So, before the following steps for APC are applied, the USPC is applied to simultaneously adjust all the transmit powers down first.

## 7. SCALABILITY OF NETWORK CAPACITY: NUMERICAL RESULTS

We generate grid topologies in a 1x1 $km^2$ domain with randomly placed clients. Initially, there are four APs with transmit power 281.8mW so that together they cover the whole domain. For 802.11, we set *VCSRange* = *PCSRange* = 2.78 x *TxRange*. We vary the number of APs in the domain while fixing the client-to-AP ratio to 5:1.

In NS-2 [5], a UDP link is established from every client to its closest AP with a data rate of 11 Mbps, and packet size of 1400 bytes. The *MAC_RTSThreshold* is 1000 bytes.

For the case of USPC, we scale the *TxRange* of APs to just cover a grid area to ensure links could be established with the minimum power. For APC, we adopt Strategy 2 as the power control scheduling strategy for our simulation here. The performance results here pertain to the power assignments after one round of the execution of the Strategy-2 APC.

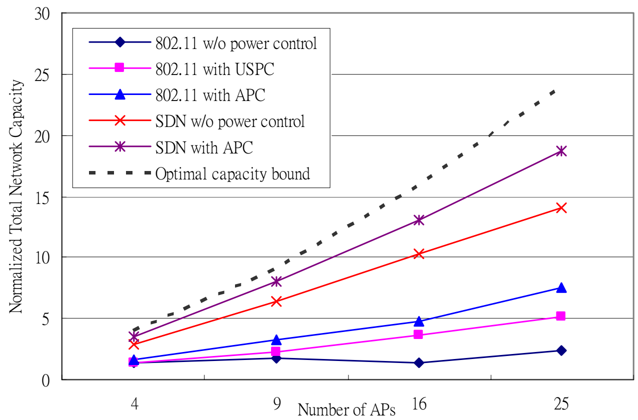


Figure 10. Simulation results of the scalability of 802.11 and SDN with and without power control in NS-2.

Figure 10 shows the simulation results in NS-2, we normalize the total throughput with the data rate as the normalized capacity. The dotted line is the maximum achievable normalized capacity in infrastructure networks, which is the total number of APs. It can be seen from the figure that the normalized capacity of 802.11 without power control saturates very quickly. Specifically, it stops at around two, while the capacities of the other four cases increase almost linearly with the number of APs.

For 802.11 with USPC, the total network capacity is around 25% of the optimal capacity, and it can be raised to about 30% with APC. It is clear that SDN performs much better than 802.11 due to the removal of extraneous carrier-sensing constraints. 56% of the optimal capacity can be achieved with SDN without power control. With APC, SDN can achieve about 76% of the optimal capacity.

## 8. CONCLUSION

This paper has investigated the capacities of 802.11 and SDN wireless networks with power control. In particular, we have considered strategies for transmit power control in which 1) the transmit powers of all nodes are uniformly scaled down by the same amount as node density increases; and 2) the transmit powers of different nodes scale down by different amounts in accordance to its interference relationships with surrounding nodes. We refer to the former as Uniformly-Scaled Power Control (USPC) and the latter as Adaptive Power Control (APC). With respect to APC, we have investigated a number of strategies, and analyzed and proved the design requirements of their various components. In addition, we have also conducted simulations to study their performances. Overall, the main contributions of this paper are three-folds:

1. An analytical framework for systematic study of various APC schemes has been provided.
2. That 802.11 networks are scalable with USPC and APC has been demonstrated.
3. However, SDN can achieve substantially higher capacity with APC; in particular, it has been shown that APC allows SDN to achieve capacity within 75% of the theoretical optimal capacity of infrastructure-mode wireless networks.

Table 1. Summary of the capacity scalability of 802.11 and SDN with and without power control based on NS-2 simulation results.

| | IEEE 802.11 | SDN |
|---|---|---|
| w/o power control | Non-scalable | ~56% scalable |
| with USPC | ~25% scalable | ~56% scalable |
| with APC | ~ 30% scalable | ~76% scalable |

Table 1 summarizes details of our findings. Our work points out that although power control can solve scalability problem, different scalable networks may have different “degrees” of scalability. Although two schemes may both have network capacities that scale linearly with the number of APs, the “slope” of the linear curve may be different.

The ratio of the slope of the curve to the slope of the optimal curve gives the degree of optimality, wherein the optimal normalized capacity is defined to be the number of APs. Thus, using the NS-2 simulation results as benchmarks, we may say that 802.11 with USPC is 25% scalable, with APC is 30% scalable; while SDN with APC is 76% scalable. This scalability measure may serve as a metric for future studies of other network variations, such as networks with directional antenna.

## REFERENCES


[1] P. Gupta, P. R. Kumar, “The Capacity of Wireless Network”, IEEE Trans. on Information Theory, Vol. 46, No. 2, pp.388-404, Mar. 2000.

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

[3] E. Jung, N. Vaidya, “A Power Control MAC Protocol for Ad Hoc networks”, Mobicom 2002.

[4] 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 Workshop, May 2003.

[5] “The Network Simulator – NS-2”, http://www.isi.edu/nsnam/ns.

[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] V. Kawadia and P. R. Kumar, “Principles and Protocols for Power Control in Ad Hoc Networks”, IEEE J-SAC, pp. 76-88, vol.23, no. 5, Jan. 2005.

[8] J. P. Monks, V. Bharghavan, and W. Hwu, “A Power Controlled Multiple Access Protocol for Wireless Packet Networks”, IEEE INFOCOM 2001.

[9] A. Agarwal, P. R. Kumar, “Improved Capacity Bounds for Wireless Networks”, Wireless Communications and Mobile Computing, vol. 4, pp. 251-261, 2004.