The COMPOW protocol for power control in ad hoc networks: Theory, architecture, algorithm, implementation, and experimentation*†

Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas
and
P. R. Kumar

Dept. of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois
1308 West Main Street
Urbana, IL 61801
USA

Abstract

We provide a new protocol for the power control problem in ad hoc networks. We describe the issues in conceptualizing the power control problem and provide an architecturally simple as well theoretically well founded solution. The solution is shown to simultaneously satisfy the three objectives of maximizing the traffic carrying capacity of the entire network, extending battery life through providing low power routes, and reducing the contention at the MAC layer. Further, the power control protocol hs the plug and play feature that it can be employed in conjunction with any routing table based routing protocol, and provides low power routes with any such protocol. The protocol has been implemented in the Linux kernel. We describe the software aspects, as well as the results of the experiments conducted.

1 Introduction

Power control is important in wireless ad hoc networks for at least two reasons: (i) It can impact on battery life, and (ii) It can impact on the traffic carrying capacity of the network.

*This material is based upon work partially supported by the U.S. Army Research Office under Contract DAAD19-00-1-0466, and the Office of Naval Research under Contract N00014-99-1-0696. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

†Please address all correspondence to the last author at University of Illinois, Coordinated Science Laboratory, 1308 West Main Street, Urbana, Illinois 61801, USA. Email: prkumar@uiuc.edu. Web: black.csl.uiuc.edu/∼prkumar.

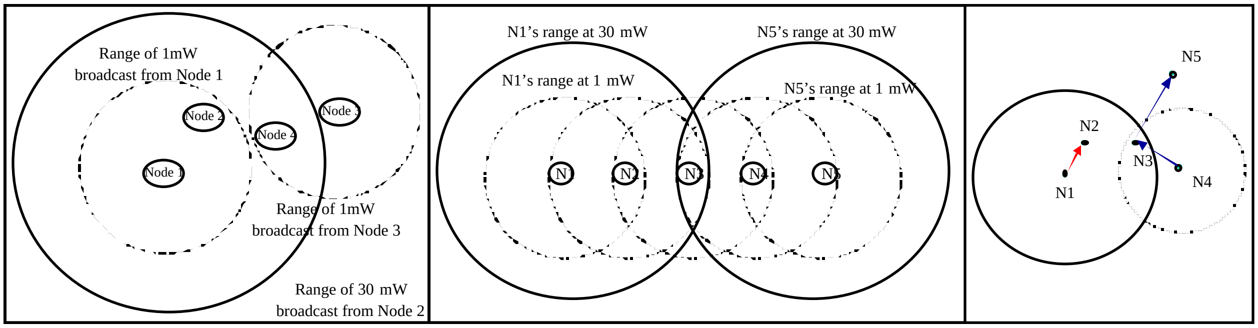

Figure 1:
Left: The dashed circle around a node indicates its range at 1mW, while the solid circle indicates its range at 30mW. Node 2 can receive a packet from Node 1 whether Node 1 transmits at 1mW or at 30mW. If Node 1 transmits at 1mW to Node 2 at the same time that Node 3 transmits to Node 4 at 1mW, then both transmissions are successful. However, if Node 1 transmits at 30mW, then Node 4 cannot successfully receive from Node 3.
Center: If all nodes transmit at 1mW, then the route from node $N1$ to $N5$ is $N1 \rightarrow N2 \rightarrow N3 \rightarrow N4 \rightarrow N5$. However, if they all transmit at 30mW, then the route from $N1$ to $N5$ is $N1 \rightarrow N3 \rightarrow N5$.
Right: $N1$ broadcasts intermittently to $N2$ at a high power level which is also heard at $N3$, then $N4$ which is sending packets to $N3$ experiences a loss of several packets which collide with the transmission from $N1$. This affects the transport layer overseeing packets from $N4$ to $N3$ to $N5$.

For the first point, note that there is no need for Node 1 in the left diagram in Figure 1 to transmit at 30mW to send a packet to the neighboring Node 2, since Node 2 is within range even at 1mW. Thus it can save on battery power. For the second point, suppose that in the same figure, Node 3 also wishes to broadcast a packet at the same time to Node 4 at 1mW. If Node 1 broadcasts at 1mW to Node 2, then both transmissions can be successfully received simultaneously, since neither is Node 2 in the range of the interferer (for its reception from Node 1) Node 3, nor is Node 4 in the range of the interferer Node 1. However, if Node 1 broadcasts at 30mW, then that interferes with Node 4's reception, and so only one packet, from Node 1 to Node 2 is successfully transmitted. The necessity for power control is thus clear. One wants an adaptive choice of power level by nodes in the network, which is implementable in a distributed asynchronous fashion by the nodes participating in the network.

The next issue that arises is: Where in the layered hierarchy does power control for ad hoc networks fit in? The difficulty is that it infringes on several layers. Clearly, power control impacts on the physical layer due to the need for maintaining link quality. However, power control also impacts on the network layer, as shown in th center diagram in Figure 1. If all nodes are transmitting at 1mW, then the route from Node $N1$ to node $N5$ is $N1 \rightarrow N2 \rightarrow N3 \rightarrow N4 \rightarrow N5$. However, if they all transmit at 30mW, then one can choose the route $N1 \rightarrow N3 \rightarrow N5$.

In addition, power control also impacts on the transport layer. In the right diagram in Figure 1, every time node $N1$ transmits at high power to node $N2$, it causes interference at $N3$ to the packets from $N4$. Thus there is a loss of several such packets on the link from $N4$ to $N3$. This impacts on the congestion control algorithm regulating the flow from source $N4$ to destination $N5$ via the intermediate relay node $N3$.

Since the success of the internet is due in significant part to the plug and play feature arising from the notion of peer-to-peer protocols which reside at well identified layers, we see that power control needs a proper conceptualization, and a properly modular solution, if the proposed solution is to be accepted.

In this paper we propose such a conceptualization, and a natural solution. The complete protocol, called COMPOW, has been implemented, and we report on the experimentation too.

The outline of this paper, and indeed the argument leading up to our solution, is organized as follows. First we argue that, for simplicity of operation, it is advantageous to have bidirectional links. Then we argue that such bidirectional links result when all nodes operate at the same power level. The next issue is what this common power level should be. We argue that it should be the smallest common power level which results in connectivity of the overall network, i.e., all nodes lying in one connected component. We show that the three objectives of enhancing the traffic carrying capacity of the network, extending battery life through power aware routing, and reducing contention at the MAC layer, are all are met through such a strategy. This also shows that power control should be situated in the network layer. We then show how to determine this lowest common power level which results in network connectivity. Now the issue becomes architectural. How does one propose a plug and play modular approach for power control? We propose a simple and natural solution using parallel modularity at the network layer. The number of nodes which can be reached from a given node can be determined from the entries in a routing table. Thus we simply run several routing daemons in parallel, each at a different power level. Comparing the entries in a routing table allows a node to choose the smallest power level at which the maximal number of nodes can be reached. This comparison is done through a simple switching logic in a distributed asynchronous manner at the nodes, another important attribute of the solution, and provides the routing table to be followed by DATA packets. The routing tables at various power levels are maintained by exploiting the port demultiplexing feature of UDP for the control packets. Next arises the issue of optimizing to reduce latency of switching from one power level to another. With current off-the-shelf cards, this latency is very large. To optimize this, we "serve" packets in a round robin clearing (or exhaustive service) fashion to eliminate needless switches of power level. The end result is the algorithm COMPOW, which has been fully implemented. It provides a joint routing and power control solution. Any routing table driven routing algorithm can be employed (the plug and play feature at work); we have chosen DSDV for simplicity. We also exhibit some experimental results.

Now for a brief survey of the literature. Current work can be loosely classified into three categories. The first class comprises strategies to find an optimal transmit power to control the connectivity properties of the network or a part of it, which could be power per node, per link, or a single power level for the whole network. [8] proposes that each node adjusts its transmit power so that its degree (number of one-hop neighbors) is bounded. [9] proposes a distributed topology control algorithm using direction information where a node grows its transmit power until it finds a neighbor in every cone of angle α, where $\alpha \leq 2\pi/3$. [11] proposes using transmit power control to optimize the average end-to-end network throughput by controlling its degree. [12] proposes an adaptive clustering scheme using transmit power control. The second class of approaches could be called power aware routing. Most schemes use the distributed Bellman-Ford algorithm with power as the metric. Some suggestions in [13] include energy consumed per packet, time to network partition, variance in node power levels, cost per packet, and node cost. [14] proposes using physical and link layer statistics to compute the power cost of a link. [10] takes the cost as the power required to reach the destination calculated by using 5m resolution GPS exchanged through flooding. The third class of approaches aim at modifying the MAC layer. [15] suggest modifying IEEE 802.11's handshaking procedure so that nodes are allowed to transmit at a low power level. [16] proposes enabling nodes to power themselves off, when they are not actively transmitting or receiving.

Our work provides an architecture for a modular implementation, guarantees bidirectionality of links, connectivity of the network, asymptotically maximizes the traffic carrying capacity, provides power aware routes, reduces MAC contention, and can even be used with any routing table driven protocol. To the best of our knowledge, it is the only scheme which has been implemented and tested on a real wireless testbed.

2 Bidirectional links are good

We begin by arguing that it is good to have bidirectional links. First note that the wireless medium is inherently lossy due to obstructions, shadowing, multipath effects, fading, etc., in addition to interference from other nodes. So, even on a single hop, a packet can never be assumed to have been successfully received by a neighbor unless the neighbor acknowledges it. (We believe that link level acknowledgments are a must in ad hoc networks). However, if an acknowledgment from a receiver R is sent at a lower power than that used for the packet transmitted by T to R, then the ACK may not be heard by R, as shown in the diagram in the left in Figure 2. Thus the link from T to R needs to be bidirectional.

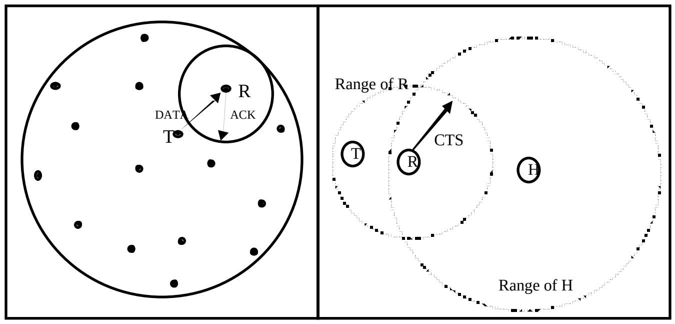

Figure 2.
Left: If the power level employed by R is lower than that employed by T, then the link (T, R) is not bidirectional. Then R's ACK may not reach T which sent the DATA packet.
Right: R is in range of H, but H is not in range of R. R's CTS is not heard by H, and H is not silenced in IEEE 802.11. So H may transmit while R is receiving a packet from T and thus cause a collision at R.

Bidirectional links are also needed for the proper functioning of a MAC protocol such as IEEE 802.11 [1, 2]. Suppose H's power level in the center diagram in Figure 2 is large enough that R is in its range. Then R needs to silence H in order to receive a packet from T. However, if R broadcasts at a lower power than H, which is insufficient to reach H, then H is not silenced. Hence H may broadcast while R is subsequently receiving a DATA packet from T, causing a collision.

Bidirectional links are also important in the network layer. When using an algorithm such as distributed Bellman-Ford, bidirectionality of links is implicitly assumed. links are bidirectional, a destination node D can send an end-to-end ACK to a source node S by merely reversing the links on the forward path taken by the DATA packet from S to D. However, if, as in diagram in the right in Figure 2, links are not bidirectional, then the ACKs have to follow different paths.

For all these reasons, as well as for supporting several other features in the TCP/IP stack present in most hosts, such as ARP, RARP, etc., it is important to have bidirectional links.

3 Common power ensures bidirectionality

How can bidirectional links be ensured? The simplest way, assuming nodes are homogeneous, is for nodes to transmit at the same power. Since all physical paths taken by radio waves from a node N to a node M can be reversed, be they multipath or reflection, and the attenuation is the same in either direction, it follows that if two nodes N and M transmit at the same power, then if M can hear N, it will follow that N can also hear M. (Note that this does not require a spherical reception region for the range). Thus we propose the use of a common power by all nodes to ensure bidirectionality of links.

4 Common power asymptotically optimizes the traffic carrying capacity

A question naturally arises: How much of a network's potential traffic carrying capacity is sacrificed by insisting that all nodes operate at a common power level? The answer is "not much," as the number of nodes is increased. This answer comes from [3] where it is shown that (under the Protocol Model) the per node throughput for a random destination can never be more than $\frac{c}{\sqrt{n}}$ for every n, where n is the number of nodes in the network, even if all transmissions are allowed to be at different power levels. However, a per node throughput of $O\left(\frac{1}{\sqrt{n \log n}}\right)$ can be guaranteed even in a network with randomly located nodes and even when all nodes broadcast at a common power level. The additional factor $\frac{1}{\sqrt{\log n}}$ is negligible.

5 What should the common power level be?

What then should this common power level be? To understand the crux of the matter, it is essential to see what the consequence is of choosing too high a power level, and of choosing it too low.

To understand the consequence of choosing too high a power level, it is essential to recognize the key property of the wireless medium: It is a shared medium. Thus, choosing an excessively high power level causes excessive interference. This reduces the traffic carrying capacity of the network in addition to reducing battery life. Consider Figure 3. On the left are shown the links that result in the network when the power level is high. The large number of links connected to node R means that whenever it is receiving a packet, five out of the six nodes A, B, C, D, F, and G have to stay silent. On the other hand, in the network on the right, the smaller power level results in fewer links, and so whenever R is receiving, only three out of the four nodes A, B, F, and G are disabled for transmitting. As will be shown more quantitatively in the sequel, the excessively large power level decreases the traffic carrying capacity of the network.

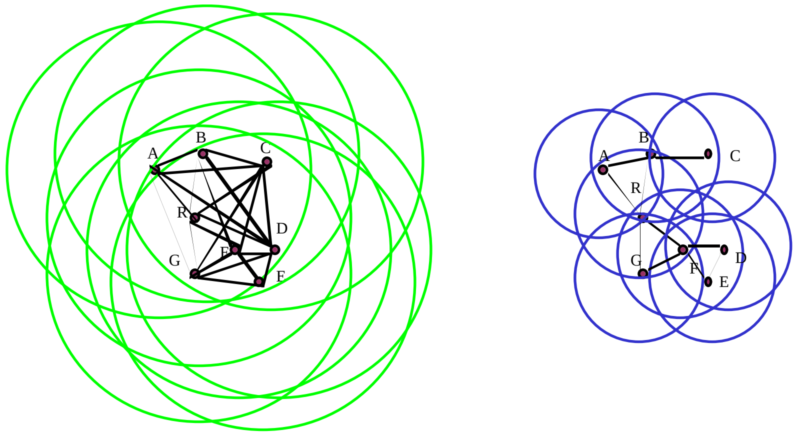

Figure 3: When power levels are too high, as in the network on the left, there are too many links, in contrast to the network on the right which results from a smaller power level. Each link means that when a node is receiving, of all the nodes connected to it by a link, only one can transmit to it. Thus, in the network on the left, when R is receiving, only one out of A, B, C, D, F, or G, can transmit. However, in the network on the right, when R is receiving, only three out of A, B, F, or G, are disallowed to transmit.

The key consequence of choosing too low a power level is that the network can get disconnected, i.e., partitioned. In Figure 4, the network on the left results from using too low a power level. As can be seen, the network is disconnected. However in the network on the right, the power level is higher and results in a set of links adequate to provide a connected network.

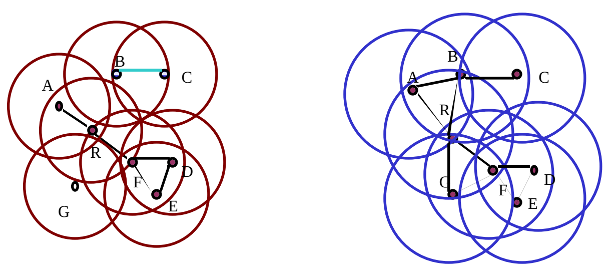

Figure 4: In the network on the left, the power level is too low. Thus the network is partitioned into three disconnected components $\{B, C\}$, $\{A, R, D, E, F\}$ and $\{G\}$. In contrast, in the network on the right, a larger power level results in a connected network.

The above qualitative discussion shows that the network power level must be chosen neither too large to cause excessive interference which results in a reduced ability to carry traffic, nor too low to result in a disconnected network.

We will now show through a quantitative argument, simplified from [3], that the traffic carrying capacity is maximized when the range is chosen to be as small as possible while preserving connectivity.

Consider a domain of area A square meters, taken as a disk for simplicity of discussion, and containing n nodes, as shown in Figure 5.

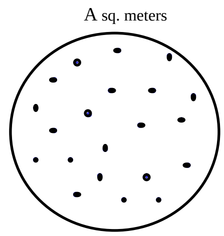

Figure 5: A disk of area A square meters containing n nodes.

Suppose that each node can transmit at W bits/sec, and that the range of each node is R meters. To model interference, let us simply suppose that for a node R to successfully receive a packet from node T, it has to lie within a distance r from R, and there can be no other simultaneous transmitter within a distance $(1+\Delta)r$ of R. The quantity r captures the range, while the quantity $\Delta > 0$ captures the notion of allowing only weak interference.

Suppose each source node has a destination node to which it wishes to send data at a rate of λ bits/sec. Suppose also that the average distance between a source and a destination is L meters. The question we will investigate is: How does λ depend on r?

For simplicity, assume a slotted operation (otherwise the capacity is even less). The average number of hops per source-destination pair is at least $\frac{L}{r}$. Thus the n nodes, in totality, require $\frac{Ln}{r}$ hops. Each hop requires λ bits/sec. Hence a total of at least $\frac{Ln\lambda}{n}$ bits/sec *needs to be transmitted by all the transmitters on average* in order to carry the per node throughput of λ bits/sec.

Now let us examine how much can actually be transmitted. Consider two simultaneous transmissions, one from T to R, and another from T' to R', as shown on the left in Figure 6. For R' to hear T', we need $|T'-R'| \leq r$ (where $|T'-R'|$ the denote distance between T' and R'). On the other hand, to avoid interference we need $|T'-R| \geq (1+\Delta)r$. From the triangle inequality, we see that $|T'-R'|+|R'-R| \geq |T'-R| \geq (1+\Delta)r$. Hence $|R'-R| \geq (1+\Delta)r - |T'-R'| \quad \geq \quad (1+\Delta)r - r = \Delta r$. Thus, disks of radius $\frac{\Delta r}{2}$ around R and R' are disjoint, as shown on the right in Figure 6.

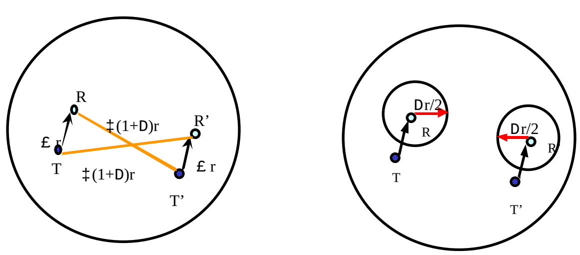

Figure 6: Two simultaneous transmissions are in effect (on the left). For successful reception of both packets, we need $|T-R| \leq r$, $|T'-R'| \leq r$, $|T'-R| \geq (1+\Delta)r$, $|T-R'| \geq (1+\Delta)r$. Hence disks of radii $\frac{\Delta r}{2}$ around receivers are disjoint (on the right).

The interpretation of this is that each transmission "consumes" a "wireless footprint" of area $\frac{\pi\Delta^2 r^2}{4}$. Thus, we observe another important fact: *Area is a valuable resource too in ad hoc networks*, in addition to the shared wireless spectrum.

Note that the total area of the domain is A square meters. Each transmission consumes an area $\frac{\pi\Delta^2 r^2}{4}$, of which at least a fourth must lie in the domain even if the receiver is on the boundary of the domain. Thus, at most $A/\frac{\pi\Delta^2 r^2}{16} = \frac{16A}{\pi\Delta^2 r^2}$ transmissions are simultaneously feasible. Each transmission can be at W bits/sec. Hence the total number of bits/sec that *can be transmitted is no more than* $\frac{16A}{\pi\Delta^2 r^2}$.

Comparing the at least $\frac{Ln\lambda}{r}$ bits/sec needing to be transmitted with the at most $\frac{16A}{\pi\Delta^2 r^2}$ bits/sec that can be transmitted, we see that in order for a per node throughput of λ bits/sec, we need $\frac{Ln\lambda}{r} \leq \frac{16A}{\pi\Delta^2 r^2}$, i.e.,

$$\lambda \leq \frac{16A}{\pi\Delta^2 L} \cdot \frac{1}{nr} \text{ bits/sec.}$$

Due to the reciprocal dependence of the right hand side on r, one wishes to decrease r. However, too low a value of r results in network disconnectivity. This justifies our goal of reducing the common power level to the lowest value at which the network is connected.

What is the smallest value of r at which the network is connected? This has been rigorously studied in [4]. Consider a network with n nodes randomly (i.e., uniformly and independently distributed) placed in a domain of area A square meters as above. Suppose $r(n)$ is the common range employed when there are n nodes. Then $\lim_{n\to+\infty}$ Prob (Network with common range is connected $\to 1$, if and only if $r(n) = \sqrt{\frac{A\ln n+\gamma(n)}{n\pi}}$, where $\gamma(n) \to \infty$ as $n \to +\infty$. This shows that the critical range for connectivity of n randomly placed nodes in A square meters is $(1+\epsilon)\sqrt{\frac{A\ln n}{n\pi}}$, where $\epsilon > 0$.

This formula provides a guideline for choosing the range, and can be used in design decisions for ad hoc networks. From the above critical radius one can also see that $\lambda(n) \leq \frac{16\sqrt{A}}{\sqrt{\pi}\Delta^2 L}\frac{1}{\sqrt{n\ln n}}$bit/sec in a network with n randomly located notes, under the protocol mode of interference. See [3]. The consequence that $\lambda(n) = O\left(\frac{1}{\sqrt{n\ln n}}\right)$ is, also, we believe, very important for design. However, it cannot be used to set up actual power levels in practice because the domain area may be unknown, the number of nodes may be unknown, and the formula is asymptotic.

6 A low common power level also provides low power routes

In the previous section we have shown that choosing the smallest range subject to maintaining network connectivity maximizes the traffic carrying capacity. However, power control also impacts on battery life. Thus the question arises: Does choosing a low power level also improve battery life? That is, is power control for the purpose of increasing traffic carrying capacity in conflict with "power aware routing?" Fortunately, the answer is "no." Not only is traffic carrying capacity maximized, but power per route is also reduced when power levels are chosen low, as the following argument shows.

Let us suppose that path loss in the medium follows an inverse α-th law with $\alpha \geq 2$, i.e., the received power at a distance ρ from a transmitter using a power level P_{trans} is $\frac{cP_{\text{trans}}}{\rho^\alpha}$, where c is a constant. Suppose that in order to receive a packet the received power level must be at least γ, i.e., $\frac{cP_{\text{trans}}}{\rho^\alpha} \geq \gamma$. Then the needed transmitter power level is at least $\frac{\gamma\rho^\alpha}{c}$. Thus, if a route from a source to a destination consists of h hops, of distances ρ_i, $i = 1, 2, \ldots, h$, then the power cost of the route is $\frac{\gamma}{c}\sum_{i=1}^{h} \rho_i^\alpha$. We can ignore the scaling variable and just fix the power cost of the route to be $\sum_{i=1}^{h} \rho_i^\alpha$.

Now consider a planar domain within which are n nodes at locations $X_1, X_2, \ldots, X_n$. For a given source node X_s and destination node X_d, let path $p = (X_0 = X_s, X_1, X_2, \ldots, X_{h-1}, X_h = X_d)$ be a *power optimal path* if it minimizes $\sum_{i=1}^{h}(X_i - X_{i-1})^\alpha$ over all $X_1, \ldots, X_{h-1}$ and all h.

Consider now a graph G formed only from edges which lie along some power optimal path from some source node X_s, $1 \leq s \leq n$, to some destination node X_d, $1 \leq d \leq n$ and such that G provides a power optimal route between any two nodes.

Lemma. *For every $\alpha \geq 2$, the graph G can be chosen as a planar graph with straight line edges, i.e., it can be embedded in the plane such that all edges are straight lines and no two edges cross each other. The graph for any $\alpha > 2$ can be chosen as a subgraph of that for $\alpha = 2$.*

Proof: First consider the case $\alpha = 2$. Suppose to the contrary that there are edges (X_i, X_j) and (X_k, X_ℓ) which cross, as in Figure 7. Then since (X_i, X_j) is power optimal, $(X_i - X_\ell)^2 + (X_\ell - X_j)^2 \geq (X_i - X_j)^2$. If equality holds above, then we can delete the edge (X_i, X_j) and replace it with the route $\{(X_i, X_\ell), (X_\ell, X_j)\}$. So without loss of generality, assume that strict inequality holds above. Then, from Euclidean geometry, $\angle A < 90°$. Similarly, $\angle B < 90°$, $\angle C < 90°$, and $\angle D = 90°$, which is of course impossible in a quadrilateral.

Now turn to the case $\alpha > 2$. We compare this with the case $\alpha = 2$, and claim that when $\alpha > 2$ no new edges need to be added (while some edges may possibly be removed). Suppose an edge (X_p, X_q) has to be added. Then if $(X_p = X_{m_0}, X_{m_1}, \ldots, X_{m_{n-1}}, X_{m_n} = X_q)$ was the earlier power optimal path for $\alpha = 2$,

$$
\begin{aligned}
|X_p - X_q|^\alpha &= (|X_p - X_q|^2)^{\frac{\alpha}{2}} \\
&\geq (\sum_{r=1}^{n} |X_{m_r} - X_{m_{r-1}}|^2)^{\alpha/2} \text{ (because } (X_p = X_{m_0}, X_{m_1}, \ldots, X_{m_{n-1}}, X_{m_n} = X_q) \text{ is optimal for } \alpha = 2) \\
&\geq \sum_{r=1}^{n} |X_{m_r} - X_{m_{r-1}}|^\alpha,
\end{aligned}
$$

which contradicts the need to add the edge (X_p, X_q). The reason for the last inequality above is that whenever $y_r \geq 0$, and $\sum_r y_r =: Y$, then $\frac{y_r}{Y} \leq 1$, and so $\sum_r (\frac{y_r}{Y})^\alpha \leq \sum_r \frac{y_r}{Y} = 1$. So $\sum_r y_r^\alpha \leq Y^\alpha = (\sum_r y_r)^\alpha$.
$\square$

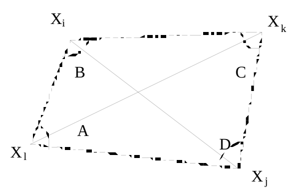

Figure 7: Suppose two edges cross.

The following simulation, replicating one in Shepard [5], was obtained by placing 500 nodes randomly in a square of side 10,000 meters. The network formed from the edges lying along power optimal routes for $\alpha = 2$ is shown on the left in Figure 8. On the right are shown the corresponding power optimal routes for $\alpha = 4$, which can be seen to be a subgraph of the graph for $\alpha = 2$.

It is clear that all hops are only to nearby neighbors, i.e., they use only low power. This illustrates that power aware routing also prefers many short hops to one long hop. Thus, low power transmission does conform with power aware routing, in addition to maximizing traffic carrying capacity. Note however that it does increase end-to-end delay.

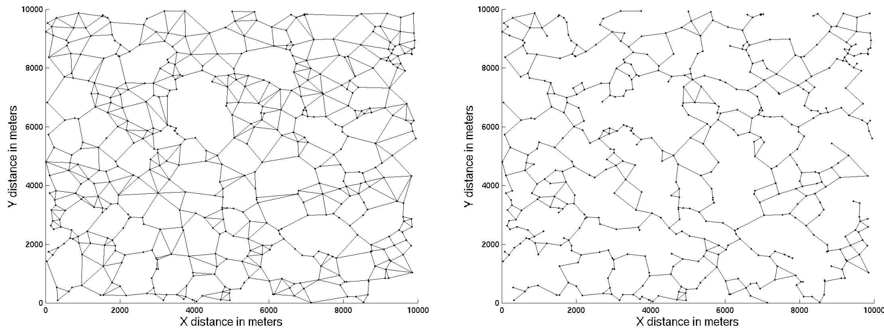

Figure 8: A network with 500 nodes randomly located in a square of side 10,000 meters. The graph on the left shows the edges lying along the power optimal paths when the path loss is $\frac{c}{\rho^2}$. The graph on the right is for $\frac{c}{\rho^4}$, and is clearly a sub-graph of the graph on the left.

7 A low power level also minimizes the contention for the MAC layer

Changing the range changes the number of neighbors that each node has, and thus the number of neighbors it has to contend with for media access. At the same time, changing the range changes the number of hops in routes, and thus the relaying burden that each node has to carry, and the amount of traffic that each node has to therefore transmit. Taking all these factors into account, does choosing a common low power reduce the contention at the MAC layer? The answer is"yes," as the following argument shows.

Suppose each node has traffic of rate λ bits/sec that it wants to to send to a destination at an average distance of L meters away. Suppose also that each node can transmit at W bits/sec, and that there are a total of n nodes randomly placed in a disk of area A square meters.

Note that the number of hops per route from a source node to its destination node is $\frac{L}{r}$ on average. Hence this is the number of relay nodes each origin-destination pair requires. Thus each node needs to transmit $\frac{L\lambda}{r}$ bits/sec on average to support the relaying burden. Now note that a node has on average $\frac{\pi r^2 n}{A}$ neighbors within a distance of r from it. These nodes are essentially colocated, and can be regarded as sharing the common channel of W bits/sec. These $\frac{\pi r^2 n}{A}$ nodes need to transmit, on average, at a total rate of $\frac{\pi n L \lambda r}{A}$ bits/sec, over a common channel which can carry W bits/sec. Since the channel of W bits/sec is fixed, the contention is reduced when the total bit-rate of $\frac{\pi n L \lambda r}{A}$ bits/sec is reduced. This clearly happens when r is reduced, i.e., at a low value of range.

Thus the MAC contention is also minimized when we choose a low common power level.

8 Power control is a network level problem

We have thus been led to the following formulation of the power control problem:

Find the smallest common power level at which the entire network is connected.

An important consequence is that since connectivity of the entire network is decided only at the network layer, *power control is properly situated as a Network Layer protocol*. Thus we have answered the conceptualization issue raised earlier concerning the layer to which power control should be relegated.

Note also that routing too is situated at the network layer. One now has to decide how network wide connectivity is to be determined at the network layer. The answer is simple for routing table driven protocols; e.g., distance vector protocols. By examining the number of nodes in the routing table for which there is a next hop, one knows how many nodes are in the connected component of the node under consideration. Thus connectivity can be judged at the network layer.

9 The need for a joint solution for power control and routing

One should note that power control impacts on the routes employed (see also the discussion on power optimal routes in Section 5). At the same time, the routing table can provide information on connectivity. Thus power control relies on the routing table, which in turn relies on power control. Clearly, the answer is to provide a joint solution for both power control and routing.

10 An architectural solution for power control: Parallel modularity

Now we turn to some architectural considerations. First note that nodes determine whether the current power level is too low or too high to give network wide connectivity and that this needs to be done in a distributed asynchronous manner by all the nodes. This means also that nodes may at any given time have different estimates of network connectivity and what power level is enough to provide such connectivity. This raises the issues of how a single node can judge connectivity at a certain power level when other nodes may possibly be employing different power levels.

Our answer to both the need for asynchronous operation, as well as the need for distributed operation, is to employ *parallel modularity* at the routing layer. We simply run several routing daemons in parallel, one for each power level.

One should note that power levels are typically discrete. For example, the CISCO 340 Series Aironet Cards allow four different power levels: 1mW, 5mW, 15mW, and 30mW, while the CISCO 350 series Aironet Cards allow six different power levels 1mW, 5mW, 20mW, 30mW, 50mW, 100mW. The number of power levels is not large, and so the number of routing daemons is feasible.

The routing tables for each power level are maintained automatically through sporadic control packets. (We have employed the DSDV protocol [7] which does feature such control packets). Thus the routing tables are operated in the standard asynchronous, distributed manner.

11 Choosing the right power level: The switching logic

Let $\{P_{min}, \ldots, P_{max}\}$ be the discretized set of power levels. Each routing table, of course, sporadically spawns its own control packets at each node to maintain routing tables for its power level. Those packets are transmitted at the power level of the particular routing table.

The issue is: What power are the DATA packets (by this we include ACK packets, and more generally, all non-routing algorithm spawned packets) transmitted at? Let $N(P_i)$ be the number of entries in the routing table corresponding to the power level P_i at a node. (By "entry" in the routing table we mean a reachable node, for example, one with a finite loop distance). Then the current power level for DATA packets is simply set to the smallest power level P_i for which $N(P_i) = N(P_{max})$, i.e., the power level $P(t)$ at time t is shown as:

$$P(t) \quad = \quad P_i \text{ if } N(P_i) = N(P_{max}) \text{ and } N(P_j) < N(P_{max}) \text{ for all } P_j < P_i.$$

Each node then chooses its power level for DATA packets based on the information contained in its own routing tables, again ensuring a distributed asynchronous operation.

Clearly, if nodes are not constantly mobile, then the routing tables for a particular power level will converge to the same set of connected nodes. Thus all nodes will settle down to the same smallest power level at which all the modes which can be connected (at P_{max}) are indeed connected.

12 Implementation: Exploiting the port demultiplexing property of UDP

How does one ensure that routing table control packets spawned by a node at a power level will be sent only to the same power level's routing table at a neighboring node? The answer is simple: We use the *port* *demultiplexing feature* of UDP.

Note first that such control packets for maintaining a routing table only traverse one hop, and are sent over UDP. Each routing table for a particular power level is implemented by a routing daemon. We simply assign a port number to each such routing daemon. The implementation taking advantage of this facility is thus as shown in Figure 9.

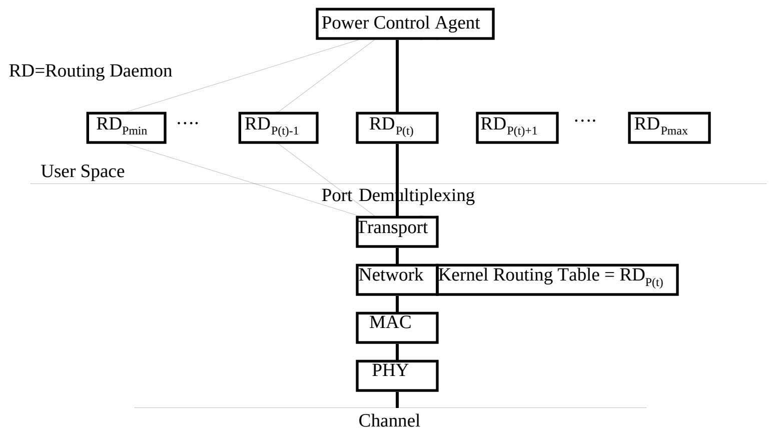

Figure 9: Outline of the COMPOW power control scheme which exploits the port demultiplexing feature of UDP.

13 Optimizing the latency due to switchover times for changing power levels

A switchover time is incurred whenever the power level at which a wireless PCMCIA card (for example CISCO's Aironet 350 card) is transmitting is changed. Measurements in the driver show that the time to change the power level in the card is about 6ms. However, when the same latency is estimated at the network layer, by monitoring the round trip time of ping packets, it turns out to be an order of magnitude higher. Thus, switching power levels too often is not desirable.

Consider the situation shown in Figure 11. If packets are sent in the FIFO order in which they arrive, as shown, then there is a large number of switchovers; in fact six switchover times are incurred to transmit the eight packets. On the other hand, if they are transmitted in the order shown on the right in Figure 11, then only two switchover times are incurred.

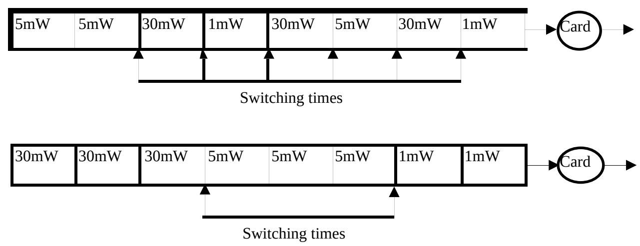

Figure 11: An excessive number of switches of power levels causes large latencies (on the top), while bunching packets of the same power level reduces the number of switchover times (on the bottom).

This problem has been studied in [6]. The goal is to bunch packets of similar power levels together, which can be accomplished by a "clearing" or "exhaustive service" policy:

> When the card is set to a certain power level, then the power level is changed only when no more packets of that power level are awaiting transmission.

The changes of the power level can then be done in a simple round-robin fashion, cycling through the power levels. This scheme is implemented between the TCP/IP stack and the device driver.

14 The software implementation of the COMPOW protocol

The implementation, done on the 2.2.16-22 Linux kernel, is shown in Figure 10. It consists of interactions between the following four entities: (i) Route daemons ("routed's"), (ii) Agent, (iii) Network driver, and(iv) Scheduler.

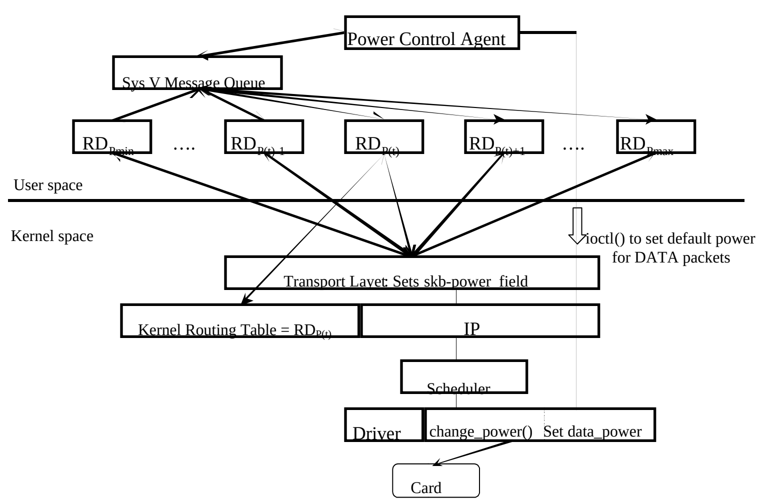

Figure 10: The software implementation of COMPOW in the Linux kernel.

Route daemons: We have chosen to use DSDV [7] for specificity as well as simplicity. We added sequence numbers to all updates, to indicate freshness and avoid loops. N of these DSDV routing daemons are run on each host, where N is the number of discrete power levels available. The input to each route daemon is the power level that it should run at. At any point there is only one "master" routing daemon, i.e., the route daemon at the current power level $P(t)$, that is allowed to modify the kernel routing table.

Agent: An agent runs on each host, which gets updates from each routing daemon of the current number of entries for that route daemon. Every update triggers a re-computation of the current power level $P(t)$ by the agent. The agent decides the new power level to be the lowest power level for which the number of entries is equal to the entries of the routing daemon at $P_{\mathrm{m}ax}$. The agent and the route daemons communicate with each other using message queues. The route daemon running at max power level is the master route daemon by default, unless instructed otherwise by the agent. There is a thread running in each route daemon that continuously monitors the message queue for messages (commands) from the agent.

Network driver: When a packet is sent down to the driver, it contains the power level that it should be sent out at, either explicitly (in control packets which are spawned by the routing daemon) or implicitly (in DATA packets). The driver sets the appropriate power level in the card and transmits the packet out on the network.

Scheduler: This component is situated between the IP layer and the network driver. The scheduler looks at the device queue before packets are pulled out of the queue and sent to the card. The packets in the queue are re-ordered to reduce the number of power changes to the minimum.

Now we turn to how the scheme is operated. When a route is added or deleted in one of the routing tables, the respective routed (route daemon) writes its power level and the new number of entries to the message queue. The agent re-computes the new power level $P(t)$ and executes the following three steps: (i) It sends a STOP command to the current "master routing daemon," i.e., the routing daemon at the current power level. (ii) It sends a START command to the new "master routing daemon." (iii) It sends down an ioctl (SIOCDATAPOWER) to the driver indicating that $P(t)$ should be the new data power level.

Since the power level at which each packet goes out on the network is "set" by the network driver, but "decided" by the route daemons and the agent, it is important that the power level information travel through the network stack hierarchy along with the packet. A new field called power_field has been added to the packet data structure (sk_buff). DATA packets have this field set to 0, but control packets have it set to the appropriate power level.

UDP is the transport layer used by the route daemons. When the route daemon sends down a packet, it sets the flags in the sendto() function call, to indicate the power level of the packet. When the packet reaches the kernel, the sk_buff (packet) structure is built for this packet. The skb→power_field is changed, based on the flags that have been set by the route daemon. Thereafter, the skb traverses IP and reaches the network device driver.

The network device driver has been changed as follows to incorporate power control. When the agent decides the current default power level $P(t)$, it is implied that all DATA packets should now use this power level. The agent does this by sending down an ioctl (SIOCDATAPOWER) to the driver, with $P(t)$ as a parameter. The driver will store the new data power level (data_power) in its private structure.

When the driver finds that there is no power level set for a packet i.e., skb→power_field = 0, as will be the case for all DATA packets, it sets the data_power on the card before sending out the packet. For a packet which has the skb→power_field set (routed packets), the driver will set the skb→power_field on the card before transmitting the packet. This will ensure that control packets go out with the respective power level of the routing daemon, and thereby the number of entries on the routing table of each power level is accurate. The driver has a new "change_power()" routine, which reads the current power level on the card, and if this power level is different from the power level of the packet that must be sent out now, then the power level of the card is changed.

15 Experimentation

The first phase of our experimentation was to develop the entire software and test it on our network of Compaq Presario 1800T laptops running Redhat Linux 7.0, kernel 2.2.16-22, with CISCO's 350 Series Aironet cards. This software development was done as described in the preceding section.

The proper functioning of the system was then tested as follows. Initially all nodes are colocated, i.e., all nodes were within range of each other at P_{min}. The power levels converge to the lowest power level 1mW. Then one node is moved away, and the power levels of all the nodes increase to a common power level of 30 mW, which is the power required to keep the network connected. Then, as the nodes are moved closer together, the power levels at all nodes reduce, converging back to 1mW when all nodes are again colocated. The results are illustrated in Figure 12.

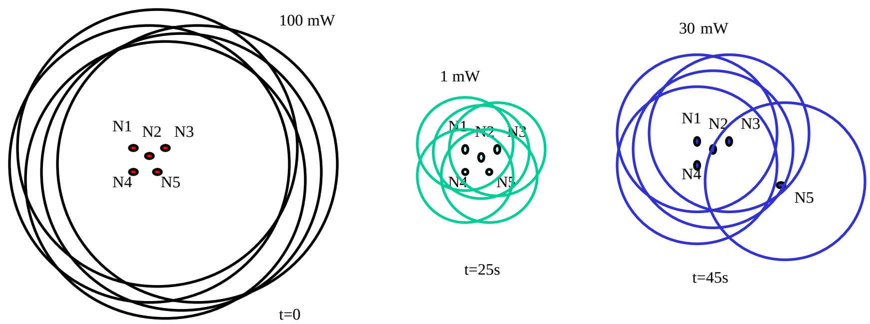

Figure 12: The converged power levels as a function of the node locations.

16 Concluding remarks

The next phase of our experiment is to study the mobility scenario thoroughly. Clearly, if the time constants of mobility are smaller than the time constants for convergence of the power levels and the routing tables, then the power control and the routing algorithms cannot keep pace with mobility. If the reverse is true with some margin, then mobility can be handled. Currently, we are studying how to reduce the latency of switching power levels, as well as optimizing parameters, to make the COMPOW algorithm as rapidly adaptive as possible. The disadvantage, but possibly a future advantage, is that our scheme is built entirely from off-the-shelf hardware. In the future we also plan to investigate the use of on-demand routing protocols, such as AODV [17], for example.

References

[1] IEEE Protocol 802.11, "Draft standard for wireless LAN: Medium access control (MAC) and physical layer (PHY) specifications." IEEE, July 1996.

[2] IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications." IEEE Standard 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[3] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 388–404, March 2000.

[4] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming* (W. M. McEneany, G. Yin, and Q. Zhang, eds.), pp. 547–566, Boston, MA: Birkhauser, March 1998.

[5] T. N. Shepard, "A channel access scheme for large dense packet radio networks," in *Proceedings of ACM SIGCOMM*, August 1996.

[6] J. R. Perkins and P. R. Kumar, “Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems,” *IEEE Transactions on Automatic Control*, vol. AC-34, pp. 139–148, February 1989.

[7] C. E. Perkins and P. R. Bhagwat, “Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers,” in *Proceedings of ACM SIGCOMM*, September 1994.

[8] R. Ramanathan and R. Rosales-Hain, ”Topology control of multihop wireless networks using transmit power adjustment,” Proceedings of *INFOCOM 2000: Nineteenth Annual Joint Conference of the IEEE Computer and Communication Societies.* vol 2, pp. 404–412, 2000.

[9] R. Wattendorfer, L. Li, P. Bahl, and Y-M. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” *INFOCOM 2001: Twentieth Annual Joint Conference of the IEEE Computer and Communication Societies*, Alaska, April 2001.

[10] V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, issue 8, pp. 1333–1344, Aug. 1999.

[11] T. A. ElBatt, S. V. Krishnamurthy, D. Connors and S. Dao, “Power management for throughput enhancement in wireless ad-hoc networks,” *IEEE International Conference on Communications.* vol. 3, pp. 1506–1513, 2000.

[12] Taek Jin Kwon and Mario Gerla, “Clustering with Power Control ,” in *Proceedings of MILCOM*, vol. 2, pp. 1424-1428, 1999.

[13] Suresh Singh, Mike Woo and C. S. Raghavendra, “Power Aware Routing in Mobile Ad Hoc Networks,” in *Proceedings of MOBICOM*, 1998.

[14] M. W. Subbarao, “Dynamic power-conscious routing for MANETs: an initial approach,” *IEEE Vehicular Technology Conference*, vol. 2, pp. 1232–1237, 1999.

[15] Jeffrey P. Monks, Vaduvur Bhargavan and Wen-Mei W.Hwu, “A power controlled multiple access protocol for wireless packet networks”, *Proceedings of INFOCOM*, vol. 1, pp. 219–228, 2001.

[16] S. Singh and C. S. Raghavendra, “Power efficient MAC protocol for multihop radio networks”, *The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, pp. 153–157, 1998.

[17] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, “draft-ietf-manet-aodv-03.txt,” Mobile Ad Hoc Networking Working Group, http://explore.kwangwoon.ac.kr/ ssbae/theory/internet-drafts/draft-ietf-manet-ao dv-03.txt, 25th June 1999.