Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol *†

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

Ph:+1 217 333 7476, Email: `prkumar@uiuc.edu`, Web: `black.csl.uiuc.edu/~prkumar`

ABSTRACT

We present a new protocol for power control in ad hoc networks. We describe the issues in conceptualizing the power control problem, and provide an architecturally simple as well as 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 protocol has the plug and play feature that it can be employed in conjunction with any routing protocol that pro-actively maintains a routing table. The protocol, called COMPOW, has been implemented in the Linux kernel and we describe the software architecture and implementation details.

1. INTRODUCTION

Transmit 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.

For the first point, note that there is no need for N1 in Figure 1 to broadcast at 30mW to send a packet to the neighboring N2, since N2 is within range even at 1mW. Thus it can save on battery power. For the second point, suppose that in the same figure, N3 also wishes to broadcast a packet at the same time to N4 at 1mW. If N1 broadcasts at 1mW to N2, then both transmissions can be successfully received simultaneously, since neither is N2 in the range of its interferer N3 (for its reception from N1), nor is N4 in the range of its interferer N1. However, if N1 broadcasts at 30mW, then that interferes with N4's reception from N3, and so only one packet, from N1 to N2, is successfully transmitted. Thus, power control can enhance the traffic carrying capacity.
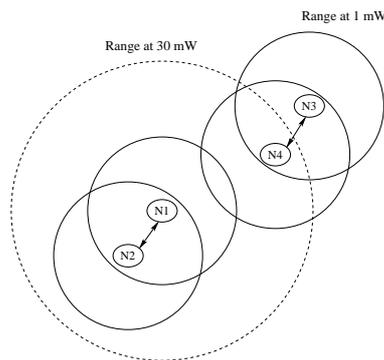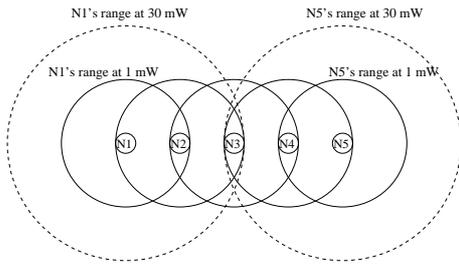
Figure 1: The need for power control

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? 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 Figure 2(a). If all nodes are transmitting at 1mW, then the route from node $N1$ to node $N5$ is $N1 \to N2 \to N3 \to N4 \to N5$. However, if they all transmit at 30mW, then one can choose the route $N1 \to N3 \to N5$. In addition, power control also impacts on the transport layer. In Figure 2(b), 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 $N3$. The need for power control is thus obvious.
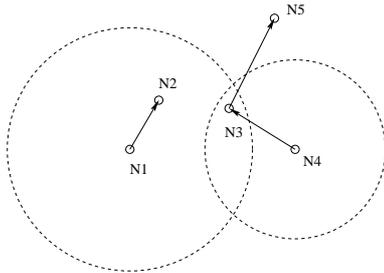
Now we provide a brief survey of the literature. Current work can be loosely classified into three categories. The first class comprises of strategies to find an optimal transmit power to control the connectivity properties of the network or a part of it, which could be power per node, per link, or a single power level for the whole network. [1] proposes that each node adjusts its transmit power so that its degree (number of one-hop neighbors) is bounded. [2] proposes a distributed topology control algorithm using direction information where a node grows

(a) Routing

(b) Transport

Figure 2: Power control affects many layers

its transmit power until it finds a neighbor in every cone of angle α, where $\alpha \leq 2\pi/3$. [3] proposes using transmit power control to optimize the average end-to-end network throughput by controlling its degree. 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 [4] include energy consumed per packet, time to network partition, variance in node power levels, cost per packet, and node cost. [5] proposes using physical and link layer statistics to compute the power cost of a link. The third class of approaches aim at modifying the MAC layer. [6] suggests modifying IEEE 802.11's handshaking procedure to allow nodes to transmit at a low power level. [7] proposes enabling nodes to power themselves off when not actively transmitting or receiving.

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. We propose just such a conceptualization, and a natural solution. This protocol aims to operate all nodes at a common power level which is chosen to be the smallest power level at which the network is connected. We provide a theoretical justification that the solution almost maximizes the traffic carrying capacity of the network, produces power aware routes, and reduces MAC layer contention. We also show how it can be implemented as a network layer protocol with plug-and-play capability respecting the IP hierarchy. The complete protocol, called COMPOW, has been implemented.

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 Figure 3(a). Thus the link from T to R needs to be bidirectional.
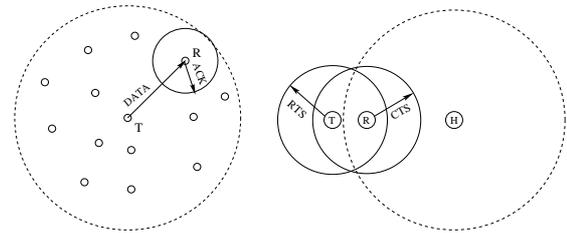
(a) ACK does not reach T
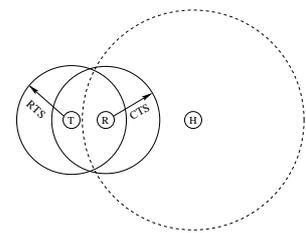
(b) CTS does not silence interferer H

Figure 3: The need for bidirectional links

Bidirectional links are also needed for the proper functioning of a MAC protocol such as IEEE 802.11 [8]. Suppose H's power level in Figure 3(b) 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 at R.

Bidirectional links are also important in the network layer. When using an algorithm such as distributed Bellman-Ford in routing algorithms, bidirectionality of links is implicitly assumed. For all these reasons, as well as for supporting several other protocols in the TCP/IP stack present in most hosts, such as ARP, RARP, etc., it is important to have bidirectional links.

3. COMMON POWER AND BIDIRECTIONALITY

How can bidirectional links be ensured? The simplest approach, 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, N can also hear M. Note that this does not require a spherical region for the range, that is, equal range in all directions.

We should note that a common power level does not guarantee a common SINR (signal to noise plus interference Ratio) at all receivers. In cellular systems, there ex-
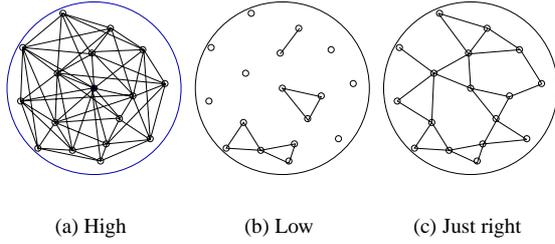
(a) High   (b) Low   (c) Just right

Figure 4: Choice of power level affects network connectivity and level of interference.

ists a feasible choice of power levels which achieve equal SINR for all the transmitters to a single receiver, which is the base station. However, in ad hoc networks, there is no choice of power levels which achieves a common SINR for all transmitter-receiver pairs. Thus the approach in the paper, as shown in the sequel, is to keep transmit powers small, hence transmit neighborhood small too, and take advantage of the spatial dispersion of nodes to achieve more or less equal SINRs.

4. COMMON POWER ASYMPTOTICALLY OPTIMIZES NETWORK CAPACITY

The question that naturally arises is: How much of a network's potential traffic carrying capacity is sacrificed by insisting that all nodes operate at a common power level, rather than selecting an optimum power for each node for each outgoing link? The answer is "not much," as the number of nodes is increased. This answer comes from [9] 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, thus showing that a common power is nearly optimal.

5. WHAT SHOULD THE COMMON POWER LEVEL BE?

A key feature of the wireless channel is that it is a shared medium. Thus, choosing an excessively high power level causes excessive interference as seen in Figure 4(a). This reduces the traffic carrying capacity of the network in addition to reducing battery life. On the other hand, in Figure 4(b), having a very small power level results in fewer links and hence network partitioning. When the power level is just right, the network is still connected and there is no excessive interference as shown in Figure 4(c).

We will now show through a quantitative argument, simplified from [9], that the traffic carrying capacity is maximized when the range is chosen to be as small as possible while preserving connectivity. Consider n nodes in a domain of area A square meters, taken as a disk for simplicity of discussion. Suppose that each node can
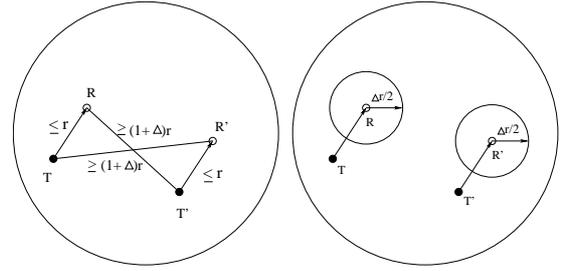
(a) Protocol model of interference   (b) Disks of radii $\Delta r/2$ around the receivers are disjoint

Figure 5: Wireless transmissions consume area

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 $\Delta > 0$ captures the notion of allowing only weak interference, i.e., capture. 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 vary with 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 $\frac{Ln\lambda}{r}$ 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 in Figure 5(a). For R' to hear T', we need $|T' - R'| \le r$ (where $|T' - R'|$ the denote distance between T' and R'). On the other hand, to avoid interference we need $|T' - R| \ge (1 + \Delta)r$. From the triangle inequality, we see that $|T' - R'| + |R' - R| \ge |T' - R| \ge (1 + \Delta)r$. Hence $|R' - R| \ge (1 + \Delta)r - |T' - R'| \ge (1 + \Delta)r - r = \Delta r$. Thus, disks of radius $\frac{\Delta r}{2}$ around R and R' are disjoint, as shown in Figure 5(b).

The interpretation 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 radio spectrum. Note that the total area of the domain is A square meters. At least a fourth of the consumed area 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* $\frac{16AW}{\pi \Delta^2 r^2}$.

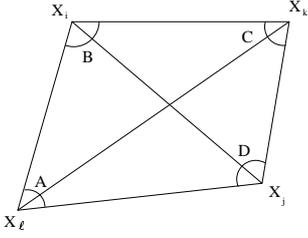Thus, we see that to sustain for a per node throughput

Figure 6: Suppose two edges cross

of λ bits/sec, we need $\frac{Ln\lambda}{r} \le \frac{16AW}{\pi\Delta^2 r^2}$, i.e.,

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

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.

6. A LOW COMMON POWER LEVEL ALSO PROVIDES LOW POWER ROUTES

Power control also impacts on battery life. Is power control for the purpose of increasing traffic carrying capacity in conflict with "power aware routing?" Fortunately, the answer is "no." Power per route is also reduced when power levels are chosen low, as shown below.

Let us suppose that path loss in the medium follows an inverse α-th law with $\alpha \ge 2$, 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., $\frac{cP_{\text{trans}}}{\rho^\alpha} \ge \gamma$. 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 $\rho_i, i = 1, 2, \ldots, h$, then the power cost of the route is $\frac{\gamma}{c}\sum_{i=1}^h \rho_i^\alpha$. We can ignore the scaling $\frac{\gamma}{c}$ 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 $X_s, 1 \le s \le n$, to some destination node $X_d, 1 \le d \le n$, and such that G provides a power optimal route between any two nodes.

Lemma: For every $\alpha \ge 2$, the graph G can be chosen as a planar graph with straight line edges, i.e. no two edges cross each other. The graph for any $\alpha > 2$ can be chosen as a sub-graph 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 6. Then since (X_i, X_j) is power optimal, $(X_i - X_\ell)^2 + (X_\ell - X_j)^2 \ge (X_i - X_j)^2$. If equality holds above, then we can delete the edge (X_i, X_j) and replace it with the route
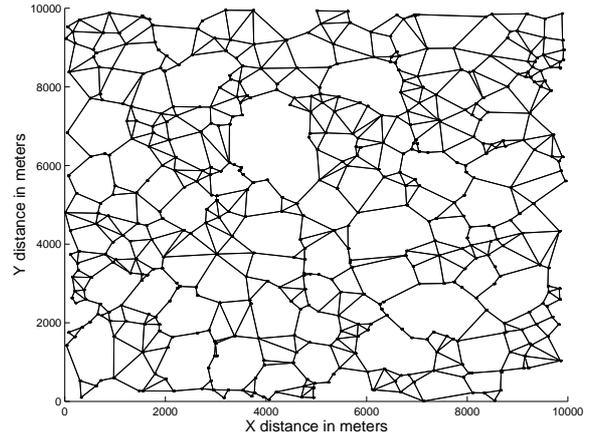

Figure 7: Graph of links lying along power optimal routes for $\alpha = 2$

$\{(X_i, X_\ell), (X_\ell, X_j)\}$. 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 some edge (X_p, X_q) has necessarily 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 $\alpha = 2$,

$$
\begin{aligned}
|X_p - X_q|^\alpha &= (|X_p - X_q|^2)^{\frac{\alpha}{2}} \quad &(1) \\
&\ge (\sum_{r=1}^n |X_{m_r} - X_{m_{r-1}}|^2)^{\alpha/2} \quad &(2) \\
&\ge \sum_{r=1}^n |X_{m_r} - X_{m_{r-1}}|^\alpha \quad &(3)
\end{aligned}
$$

which contradicts the need to add the edge (X_p, X_q). Inequality (2) follows because $(X_p = X_{m_0}, X_{m_1}, \ldots, X_{m_{n-1}}, X_{m_n} = X_q)$ is optimal for $\alpha = 2$. The reason for inequality (3) is that whenever $y_r \ge 0$, and $\sum_r y_r =: Y$, then $\frac{y_r}{Y} \le 1$, and so $\sum_r (\frac{y_r}{Y})^{\frac{\alpha}{2}} \le \sum_r \frac{y_r}{Y} = 1$. So $\sum_r y_r^{\frac{\alpha}{2}} \le Y^{\frac{\alpha}{2}} = (\sum_r y_r)^{\frac{\alpha}{2}}$. $\square$

The following simulation, motivated by one in Shephard [10], 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 in Figure 7. In Figure 8 are shown the corresponding power optimal routes for $\alpha = 4$, which can be seen to be a sub-graph of the graph for $\alpha = 2$. Due to the planarity, it can be seen that all hops are only to nearby neighbors, i.e., they use only low power, and power aware routing 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.
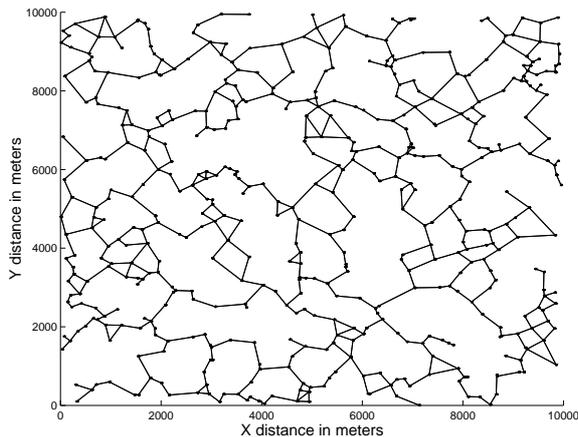
Figure 8: Graph of links lying along power optimal routes for $\alpha = 4$ is a sub-graph of the one for $\alpha = 2$

7. LOW POWER LEVEL ALSO MINIMIZES CONTENTION AT 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 consequently 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 LAYER 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*
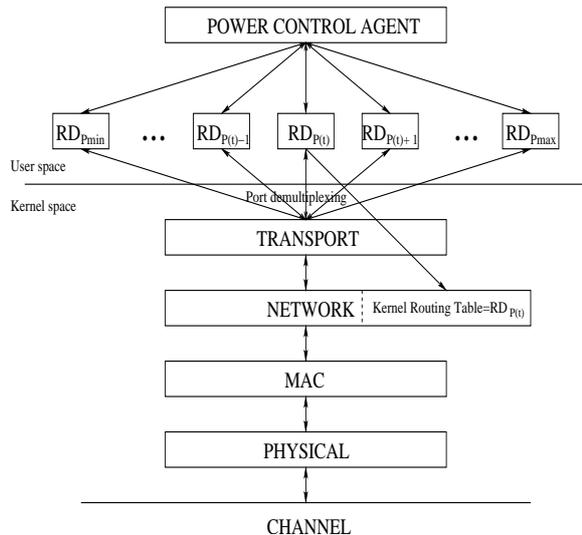

Figure 9: Architectural design of the COMPOW protocol

protocol. Thus we have answered the conceptualization issue raised earlier concerning the layer to which power control should be relegated.

Also, one of the main functions of the network layer is routing. Note that, power control impacts on the routes employed, and, vice-versa, the power control protocol needs connectivity information which is provided by the routing layer. This mutual dependency motivates the need for a joint solution for power control and routing.

9. ARCHITECTURE OF THE COMPOW PROTOCOL

The problem remaining is to design an asynchronous, distributed, and adaptive algorithm which finds the smallest common power(COMmon POWer) level at which the network is still connected. This is the essence of the COMPOW protocol.

In Section 8, we argued for a joint solution for power control and routing. The next issue we address is how to seamlessly integrate our protocol in the network stack. Our solution employs what we call *parallel modularity* at the network layer. We maintain multiple routing tables in user space, one for each of the transmit power levels available. That is, a routing table RT$_{P_i}$ for a power level P_i is constructed by sending and receiving hello messages at the power level P_i. Thus the number of entries in RT$_{P_i}$ (those nodes reachable within a finite number of hops) gives the number of reachable nodes at P_i. The optimum power level selected for the node is the smallest power level whose routing table has the same number of entries as that of the routing table at the maximum power level. If $P(t)$ is the optimum power level found, then RT$_{P(t)}$ is installed as the master routing table, which is used by the kernel. This is done by the *power control agent* which takes input from the various routing tables and decides the optimum power level. This design of the COMPOW protocol is illustrated in the Figure 9. Implementation details are described in Section 10.

We have assumed a manageable number of discrete transmit power levels in our design. This is true of the only commercially available wireless cards (at the time

of our design) which support transmit power control, namely the CISCO Aironet 340 and the 350 series. The 340 series has four power levels (1, 5, 10 and 30 mW) and 350 has six power levels (1, 5, 20, 30, 50 and 100 mW). If there are cards with many discrete power levels then we can optimize the algorithm by maintaining routing tables at power levels close to the current optimum rather than at all power levels. Of course, we always have to maintain the routing table at the maximum power level. In the event of more vendors providing different cards having different transmit ranges and power levels, there needs to be a calibration equivalence of power levels between vendors, to enable the use of diverse hardware in a network. Such consensus is required for interoperability.

The design of COMPOW employing parallel modularity fits very nicely into the OSI networking stack. We can use any routing protocol which pro-actively maintains routing tables. Since routing protocols operate in a distributed, asynchronous and adaptive fashion, COMPOW automatically inherits these features by virtue of its design. An alternative to maintaining multiple routing tables would be to use probes to flood the entire network to determine connectivity information. This would result in duplication of the work done by the routing protocol, would incur some additional overhead, would not be modular, and would result in a confusion of layers.

It might appear that running multiple routing daemons (six, in the case of Cisco Aironet 350 cards) simultaneously is a huge overhead. A simple calculation shows that it is not. Suppose each routing daemon broadcasts one hello packet of 1000 bytes every 5 seconds. This allows for about 10 entries (100 bytes per entry) in each incremental route update, which is enough for reasonable mobility rates and network size. Then, with six power levels, each node creates an overhead of 1200 bytes per second. With approximately six nodes within range which is reasonable from connectivity considerations [11], the net bandwidth consumed is 7.2K Bytes/sec or $\approx$ 60Kbits/sec. This is less than 1% of 11Mbps which is the available link bandwidth for 802.11b compliant cards (like Cisco Aironet 350). The overhead would be even less for 802.11a which supports 55Mbps. This overhead is certainly acceptable and will be easily offset by the improvement in network capacity by using a lower power level. Our preliminary experiments indicate that the overhead of maintaining routing tables is indeed negligible.

The last issue is that of the latency involved with switching power levels. Unfortunately, the switch-over latency is quite high in the CISCO Aironet cards, because they require a reset every time the transmit power level is changed, which seems wastefully unnecessary. The latency when measured in the driver was found to be 6ms, but even after the power level has been changed on the card, it takes some time to resume transmission at full throttle. When we estimated the latency of a power change at the network layer by monitoring ping traffic on the network, it was close to 100ms. However, power control is quite common in CDMA networks and is done 800 times a second. Thus, the current electronics is certainly
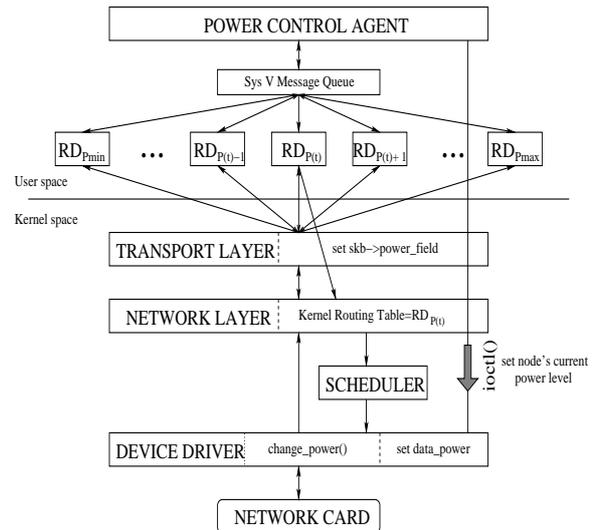

Figure 10: The software implementation of the COMPOW protocol

capable of fast power changes, but the firmware in the CISCO cards unfortunately is so written that it requires a reset for every power level change. To reduce this unfortunate switch-over latency we use a *scheduling policy*. The policy is to serve all the packets of current power level that are queued before changing the power level. In other words, bunch queued packets together according to power levels before serving them. It can be shown that this reduces the number of power level changes required [12].

10. SOFTWARE IMPLEMENTATION OF THE COMPOW PROTOCOL

A minimalist approach to the architectural design of the COMPOW protocol allows for a clean and modular implementation. The implementation illustrated in Figure 10 has been done in the 2.2.16 Linux kernel. A minimal number of changes were required in the kernel network stack and the device driver, all of them for making them aware of the concept of transmit power of a packet.

In Linux, route discovery, maintenance, etc., is done by an application, running in user space, called the routing daemon (RD). The routing daemon maintains a local copy of the routing table which it uses to update the kernel IP routing table. The kernel routing table is the one which is actually used for routing packets. So to maintain multiple routing tables at each power level, all we do is run multiple routing daemons (RD_{P_1} .... $RD_{P_{max}}$) in user space for each of the power levels available. There is a predetermined port for each power level, so that every RD_{P_i} talks only to its peers on the other nodes by virtue of transport layer demultiplexing. We use DSDV [13] as the routing protocol, which was also implemented. Now that we have routing tables at all power levels, we can decide the optimum power level for the node based on this information and update the kernel routing table. This is done by the *power control agent*, which also runs in user space, and communicates with all the routing daemons using a System V message queue. The agent continuously monitors all the routing tables and authorizes

only the routing daemon running at what is determined to be the current optimum power level to update the kernel routing table.

Since the power level at which the packet goes out is set by the network driver, but decided by the routing daemons, we need some mechanism for the power level information for each packet to travel through the kernel network stack. We achieved this by adding a field to the packet data structure (skb) to indicate the transmit power level and also adding a flag to the sendto() system call so that the desired transmit power level for each packet can be specified. Now the RD's can send hello messages (control packets) at the desired power level. The rest of the traffic (broadly called DATA packets) should be transmitted at the current optimum power level as decided by the power control agent. Thus a mechanism is needed for communicating this optimum power level from the power control agent to the network driver whenever this optimum power changes. For this we add an ioctl (SIOCDATAPOWER) to the network device driver. This ioctl sets a variable called datapower in the driver which is used for transmitting all DATA packets.

Lastly we implemented the scheduling policy described in Section 9. This was implemented in the generic dev queues between the IP and the link layer.

The entire software for the COMPOW protocol has been completely implemented and tested on our test-bed of Compaq Presario laptops running the 2.2.16 Linux Kernel and using CISCO Aironet 350 cards as the wireless interfaces.

11. CONCLUDING REMARKS

We have provided a conceptualization of the power control problem in ad hoc networks, given theoretical arguments in support of a common power level, produced a minimalist architectural design, and finally completely implemented the COMPOW protocol. Our power control protocol guarantees bidirectionality of links and connectivity of the network, asymptotically maximizes the traffic carrying capacity, provides power aware routes, reduces MAC contention, and can be used with any proactive routing protocol. To the best of our knowledge, it is the only power control scheme which has been implemented and tested on a real wireless test-bed.

The common power strategy may settle to an unnecessarily high power level when the nodes in a network are clustered. Even a single node outside a cluster can force all the nodes in the network to use a high power level. A clustering scheme is needed in such scenarios to make power control effective. In fact what we need is a joint solution for routing, clustering, and power control. Such a scheme is currently under development at the University of Illinois. We also plan to investigate the use of on-demand routing protocols like AODV. Finally, one hopes that future hardware will allow fast switching of power levels.

REFERENCES

[1] Ram Ramanathan and Regina Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of INFOCOM*, 2000, pp. 404–413.

[2] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of INFOCOM*, 2001.

[3] Tamer A. ElBatt, Srikanth V. Krishnamurthy, Dennis Connors, and Son Dao, "Power management for throughput enhancement in wireless ad-hoc networks," in *IEEE International Conference on Communications*, 2000, pp. 1506–1513.

[4] Suresh Singh, Mike Woo, and C.S. Raghavendra, "Power aware routing in mobile ad hoc networks," in *Proceedings of MOBICOM*, 1998.

[5] M. W. Subbarao, "Dynamic power-conscious routing for manets: an initial approach," in *IEEE Vehicular Technology Conference*, 1999, pp. 1232–1237.

[6] Jeffery P. Monks, Vaduvur Bhargavan, and Wen-Mei W.Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proceedings of INFOCOM*, 2001, pp. 219–228.

[7] Suresh Singh and C.S. Raghavendra, "Power efficient MAC protocol for multihop radio networks," in *The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1998, pp. 153–157.

[8] IEEE 802 LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11, 1999 edition, 1999.

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

[10] Timothy J. Shepard, "A channel access scheme for large dense packet radio networks," in *Proceedings of ACM SIGCOMM*, 1996.

[11] Piyush Gupta and P.R Kumar, "Critical power for asymptotic connectivity in wireless network," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, W.M. McEneany, G. Yin, and Q. Zhan, Eds. 1998, pp. 547–566, Birkhauser, Boston.

[12] J.R Perkins and P.R Kumar, "Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems," *IEEE Transactions on Automatic Control*, pp. 139–148, 1989.

[13] Charles E. Perkins and Pravin R. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *Proceedings of ACM SIGCOMM*, 1994.