

# On the Node-Scheduling Approach to Topology Control In Ad Hoc Networks

Budhaditya Deb

Computer Science, Rutgers University  
110 Frelinghuysen Road, Piscataway, NJ-08854  
bdeb@cs.rutgers.edu

Badri Nath

Computer Science, Rutgers University  
110 Frelinghuysen Road, Piscataway, NJ- 08854  
badri@cs.rutgers.edu

## Abstract:

In this paper, we analyze the node scheduling approach of topology control in the context of reliable packet delivery. In node scheduling, only a minimum set of nodes needed for routing purposes (usually determined by a minimum connected dominating set, MCDS) are kept active. However, a very low density resulting from switching off nodes can adversely affect the performance of data delivery due to three factors. First, our analysis shows that at low density, the average path length increases by a factor more than previously thought. Second, protocols such as the Hop-By-Hop Broadcast (HHB) reliability scheme (which relies on high network degree for optimum performance) suffer. Third, with limited buffers at nodes, the overhead is more pronounced to the extent of making the network unstable. Using probabilistic models, we derive the relationship between network density and overhead based on the above factors and find the density conditions for minimum power consumption. We also propose a, fully distributed and message-optimal node scheduling algorithm with a constant approximation bound based on the concept of Virtual Connected Dominating Sets. The scheme can asymptotically achieve optimal density conditions while adapting to different network parameters.

## Categories and Subject Descriptors

C.2.1. [Network Architecture and Design]: Network Topology

## General Terms

Algorithms, Performance, Design, Reliability, Theory

## Keywords

Ad-Hoc Networks, Topology Control, Node Scheduling, Reliable Packet Delivery, Minimum Virtual Dominating Sets

## 1. INTRODUCTION

In infrastructureless ad hoc networks with battery-powered nodes, it may be difficult and expensive to replace or recharge nodes drained of energy. Minimizing the network power consumption is not only important for maximizing the network lifetime but also for improving the maintainability of the network. In this context topology control is one of the most important mechanisms used for reducing network power consumption.

Topology control can be achieved either by transmit power control or node scheduling. In transmit power control the optimization problem is to find the transmission power (while maintaining connectivity) at which the power consumed is minimum ([1], [4], [5], [6], [7]). In node scheduling, nodes

redundant for routing are switched on and off using a scheduling algorithm ([15], [17], [18], [19], [20], [34]).

In this paper we focus on node scheduling to optimize the power consumed for *Reliable Packet Delivery* in ad hoc networks. Although similar studies have been done for the transmission power control ([4], [8], [9]), to the best of our knowledge this is the first attempt to analyze the behavior in the context of node scheduling. Most node scheduling algorithms find a minimum set of nodes required to maintain connectivity, usually with approximations to Minimum Connected Dominating Set (MCDS, [10], [11], [12], [13], [14]). MCDS minimizes the idle power dissipation and the power consumption from unnecessary reception of packets in nodes redundant for routing purposes.

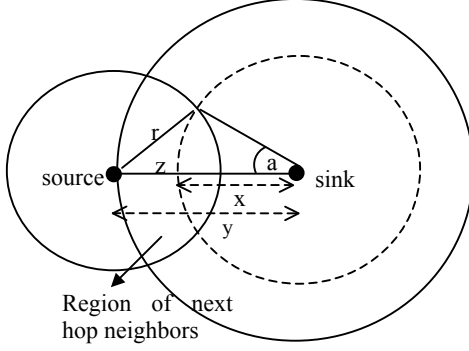
However switching off nodes can change the properties of the network graph and may actually increase the power consumption. The intuitive idea that *minimizing the network density using MCDS may not achieve minimum power consumption* forms the main motivation behind the analyses in the paper. We specifically highlight the adverse effects of keeping low network density on the overhead of reliable packet delivery and derive density conditions for scheduling nodes to minimize power consumption.

Low network density may affect overhead of reliable packet delivery in three ways. First, at low density, the average hop-length between nodes increases and may increase the power consumption. Second, many protocols such as the hop-by-hop broadcast (HHB) reliability protocol, [27] rely on high network degree (not available at low density) to reduce the overhead. Third, at low network density, the few active nodes become congestion points. If these nodes are memory constrained, this causes buffer overflow and hence increases overhead for reliable packet delivery, even leading to network instability. Thus network density plays a key role in the overhead of reliable delivery and to minimize power consumption we need to find the optimal density based on current network conditions. Further, given that we can find the optimal density we need a practical, distributed and adaptive algorithm which can achieve the required density. In this paper, we show how to find the optimal density and design a practical node-scheduling algorithm based on virtual dominating sets that achieves the desired density.

The main contributions of the paper can be summarized as follows: We first analytically derive a relationship between the density of active nodes and the distribution of shortest path lengths (section 3). The analysis shows that the expression for path lengths used in prior work ([8], [20]) is not an accurate approximation at low densities. Next, we analyze the power consumption of reliable delivery and highlight the performance of the HHB protocol at different densities in section 4.1. We analyze the effect of constrained buffers in section 4.2. The results show that operating the network at low density is counter productive. Finally we propose a practical node scheduling algorithm based on the theoretical models developed in this paper. The proposed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'05, May 25–27, 2005, Chicago, Illinois, USA.  
Copyright ACM 1-59593-004-3/05/0005...\$5.00.



**Figure 1** Computing the expected number of paths assuming a geographic forwarding scheme.

algorithm can achieve optimal density conditions in a distributed manner while being adaptive to different network conditions and parameters (section 5). Thus, the results in the paper provide design guidelines for both theoretical analysis and practical implementation on topology control using node scheduling.

## 2. NETWORK MODEL AND ASSUMPTIONS

We first describe the model and assumptions used in analysis of node scheduling. The nodes in the network are assumed to be distributed in a circular field of radius  $R$  according to a Poisson process. The methodology described can be easily extended for other network shapes and density distributions. The communication range of each node is fixed at  $r$  (no power control). We assume that the fraction of nodes kept active using some node-scheduling algorithm (possibly a connected dominating set) is also Poisson distributed (this assumption is relaxed). For a particular realization of the network, we assume that  $N$  nodes are uniformly distributed in the field and a fraction  $X$  of the  $N$  nodes form the active set. Thus we have:

$$\text{Avg. density, } \delta = NX / \pi R^2, \text{ Avg. degree} = d = \delta \pi r^2 - 1 \quad \dots 1$$

The MAC layer is assumed to be based on TDMA scheduling [26] such that collisions and interferences do not occur. Channel error is a network-wide constant ( $f$ ) and occur independently at each link. Generalization to non-uniform packet error rates (e.g. based on SNR) is part of future work.

We use a simple traffic model where nodes generate a constant number of packets per unit time ( $\lambda_p$ ) and send it randomly to some other node possibly multiple hops away from the source is. We assume a geographic routing protocol for packet forwarding.

The energy consumption in node is divided in to four categories:

$$E_T \text{ Joules} = \text{Energy consumed per Packet Transmission}$$

$$E_R \text{ Joules} = \text{Energy consumed per Packet Reception}$$

$$E_I \text{ J/second} = P_I \text{ watts} = \text{Power consumption in Idle Mode}$$

$$E_S \text{ J/second} = P_S \text{ watts} = \text{Power consumption in Sleep mode}$$

$$E_{\text{total}} \text{ J/second} = P_{\text{total}} \text{ watts} = \text{total power consumption}$$

Active nodes consume  $P_I$  at all times even when receiving or sending packets (this corresponds to the power to keep the node circuitry running). Thus, the total power dissipation *normalized with respect to the number of nodes* in the network is:

$$P_{\text{Total}} = \lambda_p (E_T O_{\text{Trans}} + E_R O_{\text{Rec}}) + P_I X + P_S (1 - X) \quad \dots 2$$

$$O_{\text{Trans}} = \text{Total number of packets transmitted.}$$

$$O_{\text{Rec}} = \text{Total number of packets received.}$$

$O_{\text{Trans}}$  denotes the expected number of packets transmitted over multiple hops including overhead due to retransmissions between a source and sink.  $O_{\text{Rec}}$  denotes the overhead when nodes in the neighborhood of the transmitting node receive all these packets.

## 3. IMPACT OF DENSITY ON SHORTEST PATH LENGTHS

In this section we study the relationship between the density of nodes active for routing and the average hop length between nodes. The results in this section highlight some of the inadequacies in papers such as [8] and [20] which approximate the hop lengths as  $\lceil y/r \rceil$  for transmission radius  $r$  and distance between nodes  $y$ . The analysis for average path lengths is similar to ones in [1] and [2] with some important differences. The derivations consider all forms of trajectories between source and sink in the probability distribution whereas in [1] and [2] the progress is assumed along a straight line joining the source and sink. We also take into account the unequal distribution of the distances between randomly chosen points in a circle (and hence unequal distribution of the hop lengths) for minimizing the power.

To derive the analytical model for the average number of hops in the network as a function of density for a pair of nodes separated by distance  $y$  we assume a geographic forwarding scheme with no holes in the network. Thus a packet progresses to the node closest to the sink. I.e., all next hop neighbors lie in the intersection of two circles of radii  $r$  (communication range) centered at source and  $y$  (the distance between source and sink) centered at the sink, as shown in Figure 1. Let

$x$  = radial distance of the forwarded node next hop and sink

$z$  = radial progress from the source towards the sink  $= y - x$

The probability that the next hop node is at a distance larger than  $z$  (i.e., the CDF) can be shown to be:

$$F_z(z) = P(Z \leq z) = 1 - \exp \left[ - \int_{t=z}^r 2a(y-t) \delta t \right] / F_z(0)$$

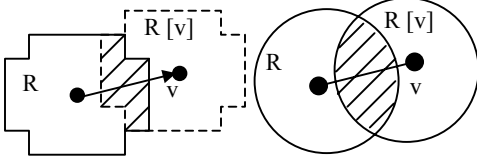
Where,  $a = \cos^{-1}(x^2 - y^2 - r^2) / 2yx$ .

$F_z(0) = 1 - \exp \left[ - \int_{t=0}^r 2a(y-t) \delta t \right]$  is the normalizing factor obtained

by putting  $F_z(0) = 1$  (to account for the assumption that there are no holes in the network and there is at least one node closer to the sink). The PDF is simply given by:

$$f_z(z) = \frac{d}{dz} F_z(z) = (2\delta a(y-z)) \exp \left[ \int_{t=z}^r 2a(y-t) \delta t \right] / F_z(0) \dots 3$$

Next we define  $h(y, z) = 1/(y-z)$ , the hops per meter progressed at distance  $y$ . To compute the number of hops from source to sink we integrate the *expected hops per meter* over the distance  $y$  from source to sink.



**Figure 2** The bounded region  $R$  shifted by vector  $v$ . The density of the parametric is proportional to the intersection area of the  $R$  and  $R[v]$ .

$$h(y) = E(h(y, z)) = \int_{h=y}^0 \int_{z=h-r}^{z=h} f_z(z)/(h-z) dz dh \quad \dots 4$$

Finally to find the average all pair shortest paths, let  $f_Y(y)$  be the PDF of the distance between two randomly chosen Poisson distributed points inside the circular field. The expected number of hops is:

$$\hat{h} = E[h(y)] = \int_{s=0}^{s=2R} \left[ \int_{h=y}^0 \int_{z=h-r}^{z=h} f_z(z)/(h-z) dz dh \right] f_Y(s) ds \quad \dots 5$$

We still need to compute the function  $f_Y(y)$  (look ahead to equation 6) in order to complete the analytical model. To derive this, consider a parametric formulation of a  $d$  dimensional bounded region  $R$  and any point defined by the vector  $v[x_1, \dots, x_d]$ . The following theorem is used to compute the density distribution of distances between randomly chosen pair of points inside a bounded region.

**THEOREM 1:** The density distribution of a pair of points with magnitude and direction given by a vector  $v$  in a bounded region  $R$  is proportional to the area formed by the intersection of  $R$  and a copy of  $R$  shifted by the vector  $v$ , ( $R[v]$ ).

**Proof:** Can be proved using simple area arguments (Figure 2).

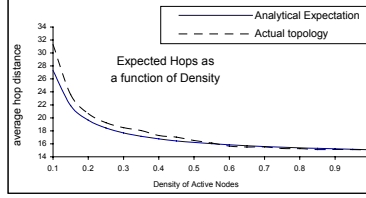
Although the theorem 1 is conceptually simple, it gives us an elegant way of handling any bounded region of different arbitrary shapes. Applying this principle to unit disks, we see that the intersection area is symmetric and equal along any direction of vector  $v$ . The overlap area of circles shifted by distance  $y$  is:

$$4 \int_{y/2}^1 \sqrt{1-x^2} dx = 2 \tan^{-1}(\sqrt{4-y^2}/y) - y\sqrt{4-y^2}/2$$

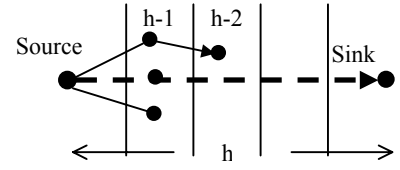
Now to find the density distribution of a pair of points with magnitude  $|v|=y$ , we use the parametric form of the vector  $v=[y\sin(t), y\cos(t)]$  and integrate along the entire circular region. In the parametric form for the distances,  $y$  is the first quadrant of the circle ( $\pi/2$ ) and the number of vectors with magnitude  $y$  is proportional to  $y$ . The integral of  $y$  from 0 to 2 is simply ( $\pi/2$ ). Hence the density distributions (PDF and CDF) is given by:

$$f_Y(y) = \frac{y}{\pi} \left[ 4 \tan^{-1}(\sqrt{4-y^2}/y) - y\sqrt{4-y^2} \right] \quad \dots 6$$

$$F_Y(y) = \int_0^y f_Y(y) dy = \sqrt{4-y^2}(-y-y^2)/2\pi + 2 \sin^{-1}(y/2)/\pi + 2y^2 \tan^{-1}(\sqrt{4-y^2}/y)/\pi$$



**Figure 3** Comparison of the equation 3 with actual hops between a pair of nodes.  $R=600m$ ,  $r=40m$ ,  $N=10000$ .



**Figure 4** The HHB scheme. Multiple Nodes at each hop distance are depicted by the partition lines. A packet travels from the  $i$ th partition the  $i$ -th partition.

We use  $f_Y(y)$  given by equation 6 in equation 5 to compute the expected number of hops as function of density. For infinite network density, the above equation would tend to the trivial approximation of hop lengths using  $\lceil y/r \rceil$ . However for networks with finite densities, the above formulation accurately estimates the path lengths. Figure 3, shows the number of hops between a pair of nodes for different densities from a randomly generated topology and compares with the analytical computation using equation 3. We see that the analytical model closely resembles the actual hops for the generated topology; thus, verifying our claim.

An important observation is that the number of hops at low densities is large and has significant cost on the network overhead. In this case of the source destinations pair, with only 10% of the nodes active, the hop lengths double and increase by 30% with around 30% of the nodes active. Impact of this significant increase in hop length cannot be neglected especially for reliable transmissions.

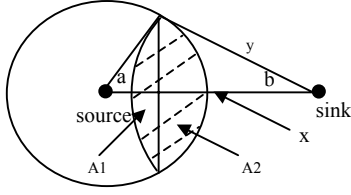
## 4. EFFECTS OF DENSITY ON RELIABLE DELIVERY OVERHEAD

Reliable packet delivery is usually achieved by retransmissions of lost or corrupted packets. At low density the number of retransmissions for reliable delivery increases not only due to increase in path lengths but also due to the following factors:

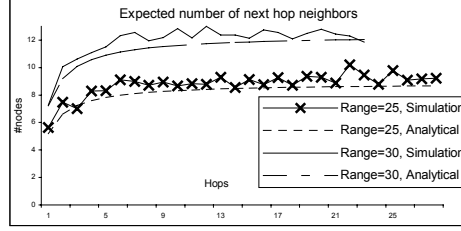
**1. Reduced network degree at low densities:** Many protocols in ad hoc networks exploit the broadcast nature of the wireless channel and high network degree to reduce their overhead (e.g., [24], [25] for routing and [26] for creating multicast trees). The Hop-by-Hop Broadcast (HHB) protocol described in [27] reduces the number of retransmissions for reliable delivery by exploiting the high network degree at high density. Low network density leads to a low network degree and may affect the performance of these protocols. This is analyzed in section 4.1.

**2. Increased traffic load in memory constrained nodes:** When very few nodes are active for routing purposes, the traffic load in these active nodes increase since a fewer number of nodes have to share entire the traffic load. With limited memory capacity, it becomes difficult to handle the increased load at low density. The overhead of reliable delivery is severely affected since buffer overflow causes extra packet losses. The effect of memory constraints on the overhead is analyzed in section 4.2.

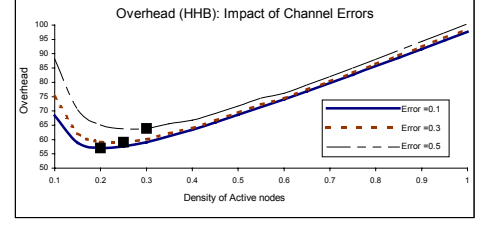
Reliable delivery protocols are broadly divided into End-to-End Retransmissions (EER) and Hop-by-Hop Retransmission (HHR) schemes. The two factors described above affects the overhead of EER and HHR, such that the optimal density for minimum power consumption is much larger than the minimum required to



**Figure 5 Computation of expected number of next hop neighbors**



**Figure 6 Comparison of equation 13 with simulation results on a real topology. R=600. N=20000**



**Figure 7 The expected power consumption using HHB. N=10<sup>4</sup>, R=600m, r=40m, E<sub>T</sub>=1J, E<sub>R</sub>=0.1J, P<sub>I</sub> = 0.01J/s, P<sub>S</sub>=0.005J/s, λ<sub>p</sub>=1**

maintain connectivity. However, detailed analysis of the normal EER and HHR schemes is left out due to lack of space.

Instead we concentrate on a concept known as Reliability with Hop-by-Hop Broadcast<sup>1</sup> (HHB) from [27] which is affected more significantly due to node scheduling. HHB significantly reduces the overhead of reliable transmissions over EER and HHR schemes by exploiting the wireless broadcast in a dense networks. In HHB, higher the local density, the lower is the retransmission overhead. Thus it is very interesting in context of this paper since, it shows that density not only affects the number of hops between nodes but also the per hop retransmission overhead.

#### 4.1 Hop-by-Hop Broadcast Reliability (HHB)

We now briefly describe the HHB protocol and analyze the effect of density on its overhead. Consider a source and sink node separated by  $h$  hops. When networks have high density, there are multiple *next-hop* nodes as shown in Figure 4. Since wireless is a broadcast medium, all the next-hop-nodes in the communication range can potentially receive a forwarded packet. This packet needs to reach at least one of these nodes so that it is forwarded with a gradient towards the sink. The probability of a packet being successfully transmitted to at least one of these nodes is higher than the probability of it being forwarded to one particular node<sup>2</sup>. The success probability of a packet getting forwarded is thus:

$$p = \text{probability to reach next hop} = 1 - f^{k(h)} \geq 1 - f$$

$$k(h) = \text{number of next-hop nodes at the } h^{\text{th}} \text{ hop.}$$

HHB exploits this inherent redundancy in the wireless medium and the independent nature of signal corruption at the receivers to reduce the per-hop retransmission overhead. Higher the network density, higher would be the number of next hop neighbors and a packet would have a higher probability to be forwarded to the next hop. This leads to lesser number of retransmissions. Intuitively HHB protocol is exactly same as a normal HHR protocol but with lesser number of retransmission at each hop. For more details about the HHB protocol please refer to [27].

Next we compute the overhead for the HHB protocol. For this we first need to compute the expected number of next hop neighbors  $k(h)$ , as a function of network density.

<sup>1</sup> Although HHB in [27] is intended to provide different levels of assurance in sensor networks, we can adapt the key ideas for a protocol employing HHB for reliable delivery of packets.

<sup>2</sup> The idea is analogous to diversity decoding with multiple wireless antennas to increase transmission reliability with the different nodes acting as multiple antennas for the same signal.

Consider a node which is  $h$  hops away from the sink as shown in Figure 5. Since we assume a geographic routing scheme, the intersection area of circles centered at the source (radius  $r$ ) and the sink (radius  $x$ ) represents the area for all  $h-1$  hop nodes. Now  $y$  is equal to  $(h-1)z$  (where  $z$  is the expected progress  $E[f_z(z)]$  per hop computed from equation 3) since we are considering the region of  $h-1$  hop neighbors. Also the source is  $h$  hops away from the sink. Thus  $x$  (as shown in figure) is given by:

$$x = z\sqrt{(h^2 + (h-1)^2)/2}$$

The area of the shaded region is  $A=A_1 + A_2$ , where,

$$A_1 = r^2[a - \sin(2a)/2], \quad A_2 = y^2[b - \sin(2b)/2] \quad \dots 7$$

$$a = \cos^{-1}(r^2 + x^2 - y^2)/2rx, \quad b = \cos^{-1}(y^2 + x^2 - r^2)/2xy \quad \dots 8$$

The expected number of next-hop nodes at the  $h^{\text{th}}$  hop is:

$$k(h) = \delta(A_1 + A_2) \quad \dots 9$$

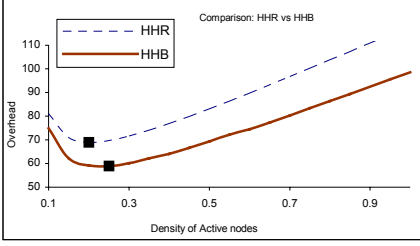
We verify the analytical expectation of  $k(i)$  using simulations for 20,000 nodes, randomly deployed in a  $1000m^2$  field for various communication ranges in Figure 6. The results suggests that equation 13 is a good approximation of  $k(h)$  and may be used effectively to compute the overhead of HHB as follows.

Next we use the analytical expression for the expected number of next hop neighbors to compute the overhead of the HHB protocol. Consider the  $i^{\text{th}}$  hop between a source and sink separated by  $h$  hops. A packet is successfully forwarded to the next hop with probability  $(1 - f^{k(i)})$  and the acknowledgement is received from the next hop with probability  $(1 - f)$ . The number of retransmissions follows a geometric probability distribution of  $(1 - f^{k(i)})(1 - f)$ . Hence the expected number of retransmissions is  $1/(1 - f^{k(i)})(1 - f)$ . The probability of an acknowledgement packet being generated for any transmission is  $(1 - f^{k(i)})$ . Hence the expected number of transmissions per packet is  $(2 - f^{k(i)})$ . The total number of packet transmissions for a pair of points separated by distance  $y$  (and hence number of hops is  $h(y)$ ) is:

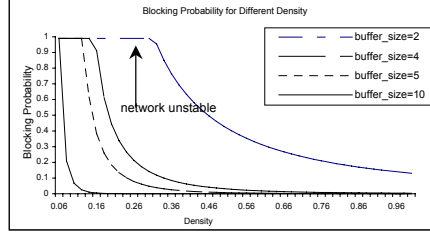
$$T_{\text{Trans}}(h(y)) = \sum_{i=1}^{[h(y)]} (2 - f^{k(i)}) / (1 - f^{k(i)})(1 - f)$$

The expected number of packet transmissions for all nodes in the network is given by equation 10 (using the equation 4, 5).

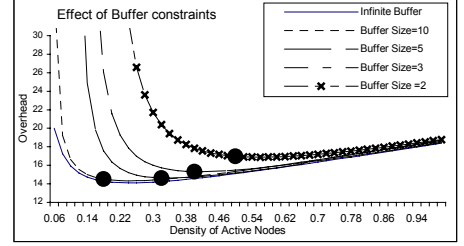
$$T_{\text{Trans}} = E[T_{\text{Trans}}(h)] = \int_{y=0}^{2R} \left[ \sum_{i=1}^{[h(y)]} (2 - f^{k(i)}) / (1 - f^{k(i)})(1 - f) \right] f_y(y) \quad \dots 10$$



**Figure 8 Power consumption for the HHR and HHB.**  $N=10^4, R=600m, f=.3, r=40m, E_T=1, E_R=0.1, P_I=0.01, P_S=10^{-3} J/s, \lambda_p=0.1$



**Figure 9 Blocking probability at different densities.**  $N=10^4, R=600m, f=.1, r=40m, E_T=1, E_R=0.1, P_I=0.01, P_S=10^{-3} J/s, \lambda_p=0.1$



**Figure 10 Effect of buffer constraints on the overhead.**  $N=10^4, R=600m, f=.1, r=40m, E_T=1, E_R=0.1, P_I=0.01, P_S=10^{-3} J/s, \lambda_p=0.1$

We have  $O_{Trans} = T_{Trans}$ ,  $O_{Rec} = T_{Trans}d$  where  $d$  is the average network degree. The expected power consumption is given by:

$$E[P_{HHB}] = \lambda_p E[O_{HHB}(h)](E_T + E_R d) + P_I X + P_S (1 - X) \quad \dots 11$$

For simple unicast HHR scheme, we can put  $f^{k(i)}=1$  in equation 10 to model that packet is forwarded to only one node at each hop.

Figure 7 plots the expected power consumption of the HHB method using equation 11. Figure 8 compares this with a hop-by-hop unicast reliability (HHR) protocol. In the graphs the optimal density point is marked with a large dot. We see that at very low and very high density the consumption is high and the optimal density point lies somewhere in the middle. The HHR and HHB plots have essentially the same curves but the HHB method has significantly lesser power consumption. We also see that increase in channel error does not have a significant impact on the overhead in the HHB case. When channel errors are high, the optimal density point for HHB lies at a much higher density than for the HHR case. This is because in HHB a high density not only lowers number of hops but also reduces the per hop retransmission overhead by exploiting the higher network degree.

## 4.2 Effect of memory constraints in nodes

We now consider the effect memory constraints on the overhead of reliable delivery at different densities. The intuitive idea in the analysis is that with lesser active nodes, the traffic rates at these active nodes increase leading to buffer overflow and subsequent packet drops. This would increase the required number of retransmissions for reliable delivery.

To find the overhead of reliable delivery in this case, we have to find the actual packet drop probability which now depends on the channel error rate, the buffer size at each node and the packet arrival rate. In a hop-by-hop reliability scheme, a node buffer gets occupied not only with new incoming packets but also with previously buffered packets which have not been reliably transmitted to the next hop. The holding time for such packets in the buffer actually depends on the rate at which packets are lost at the next hop nodes. A higher blocking probability at the next hop node thus increases the buffer occupancy at the forwarding node. Thus we need to find the buffer blocking probability at balance conditions for the network.

Let us first look at how packet arrival rates at each node increase with reduction in network density. We assume that the distribution of packet destinations is uniform and each active node equally participates in the routing. We neglect edge effects. In the HHB scheme, each forwarding node is a source for a packet on a hop-by-hop basis irrespective of the original source of the packet. When a packet is generated (or forwarded), it is intended for one

of the neighbors. Thus the average traffic volume at each node can be considered on a local basis. Based on the current network degree (from equation 1), the packet arrival rates at each node increases due to the decrease in network degree as follows.

$$\lambda_p^{in} = \lambda_p d_{actual} / d_{node-scheduled}$$

To solve the expected blocking probability at balance conditions we model the hop-by-hop retransmission scheme as a *stable* (arrival < transmission rate) *store-and-forward queuing network* ([37]) Detailed derivations are left out due to lack of space. The blocking probability,  $b$ , at each node is given by the following set of equations.

$$b = P(0) \rho^M S_M(\lambda_p^{in} T^{hold} / \rho) / \phi \quad \dots 12$$

$$P(0) = (\rho - 1) / (\rho^{M+1} S_M(\lambda_p^{in} T^{hold} / \rho) - S_M(\lambda_p^{in} T^{hold}))$$

$$\eta = (1 - \rho), \quad S_M(x) = \sum_{j=0}^M x^j / j! \quad , \quad \rho = \lambda_p^{in} (1 - f) t_{trans}$$

$M = \#$  of memory slots available for buffering packets

Now  $T^{Hold}$  is the holding time for a packet in the outgoing queue awaiting acknowledgements. Packets are buffered until they are successfully delivered to the next hop. The holding time in buffer for unacknowledged packets depends on the time required for expected number of retransmissions and one acknowledgement for a correctly received packet. If  $T^{Timeout}$  is the mean timeout interval for a packet retransmission,  $T^{Ack}$  is the mean time for the arrival of an acknowledgement and  $b$  is the blocking probability under balance conditions,  $T^{Hold}$  is given by:

$$T^{Hold} = [T^{Timeout} (1 - (1 - f)(1 - b)) / ((1 - f)(1 - b))] + T^{Ack} \quad \dots 13$$

The blocking probability,  $b$  at balanced conditions is the solution to equation 12 and 13. Figure 9 plots the average blocking probabilities at nodes for different densities and buffer sizes. We assume that  $t_{trans} = 10^{-3}$  and  $T^{Ack} = 2 t_{trans}$  and  $T^{Timeout} = 0.02$ . We see that at low densities and for small buffer sizes, the network becomes unstable as blocking probabilities reach 1.0, i.e. incoming rate is more than outgoing rate.

Finally to compute the retransmission overhead, we see that a packet is lost if there are channel errors or if the buffers are full and the packet is dropped at the receiver. The modified packet loss probability is given by equation 14 and is used for analytical computation of overhead under buffer constraints in equation 11.

$$f' = f + b - bf \quad \dots 14$$

The power consumption equation (equation 11) is plotted using equations 12, 13 and 14, in Figure 9, for different densities and buffer sizes. The optimal density points (shown by the big dots) increase as the buffer sizes reduce. Under memory constraints, the overhead at minimum density is now much higher than in the unconstrained memory case. Figure 9, also shows that network maybe even unstable at low density.

## 5. ALGORITHMS FOR NODE SCHEDULING

The results in the previous section highlight the main motivation for this paper, that many algorithms, which switch of all redundant nodes may not have overall optimal power consumption since it leads to very low network density. Our analyses suggest that optimal network density can vary widely depending on the network conditions. Thus we need an adaptive algorithm capable of scheduling nodes based on the current network conditions and parameters. The algorithm needs to maintain the following conditions on the active node set so that our analytical computations and assumptions in the previous sections can be used:

**1. Density Distribution:** The distribution of the active node set should follow the analytical assumptions (Poisson distribution). Moreover, the protocol should be adaptive to local aberrations or non-linearity in node density.

**2. Connected Dominating Set:** The active node set needs to be connected so that each node is reachable.

In general the two conditions are difficult to achieve in conjunction for a real network. We have to relax one condition in order to guarantee the other one. For routing purposes, meeting the connectivity condition proves far more important while relaxing the first condition. In section A we propose an algorithm based on Voronoi aggregative processes (VAP), which guarantees the first condition while relaxing the connectivity condition. In section B, we propose an algorithm based on the concept of Minimum Virtual Dominating Sets (MVDS) which guarantees a connected set of nodes, while relaxing the distribution condition.

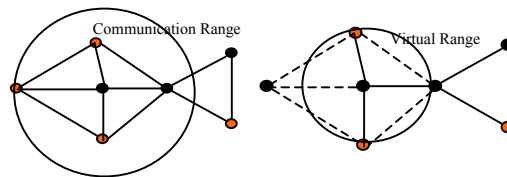
### 5.1 Constructing Poisson Distributed Active Node Set (Voronoi Aggregative Processes)

In this section we describe how the active node set can be chosen with a Poisson distribution at a desired intensity while generating a dominating set with high probability. The results discussed in this section are useful to compare the properties of our proposed algorithm in section B.

We consider two independent Poisson processes for the distribution of the *dominating* node set and the *dominated* node set forming a two level structure. For a particular realization of the network, we have  $N$  nodes uniformly distributed in the field and  $XN$  chosen randomly to form the active node set as follows.

First, a fraction  $X$  of the  $N$  nodes are chosen randomly to form the active node set. Nodes which are not chosen, associate themselves with the closest active neighbor (within one hop). Next, nodes that remain uncovered by the set of active nodes are also added to the active node set.

Let the fraction of nodes which were chosen at the 2<sup>nd</sup> step be  $Y$ . At the end of the algorithm, we have  $N(X+Y)$  nodes in the active set. The dominated nodes associate with the closest active neighbor creating a Voronoi tessellation with the first set forming



**Figure 11 MVDS (the black nodes) at different virtual ranges.**

the centers of the Voronoi cells. This is referred to as a Voronoi Aggregative Process related to Poisson distributions [21].

Now we started with a desired intensity level of  $X$  for the active node set and ended with an intensity level of  $X+Y$ . Ideally we want the achieved intensity to be equal to the desired intensity. We see that a node is chosen in the second stage if the distance to the closest Voronoi center is greater than the communication range. If the communication radius is  $r$ , the probability  $p(r)$  that a node is covered by at least one dominating node is given by:

$$p(r(Y, X)) = \exp(-XNr^2 / R^2) \quad \dots 15$$

Equating  $Y=p(r)$  and solving for  $r(X, Y)$  we get the required range.

$$r(Y, X) = \sqrt{-\ln(Y)R^2 / XN} \quad \dots 16$$

Thus if we select a small value of  $Y$ , then for the communication range given by the above equation, we get a dominating set structure with density very close to  $X$ . To create the active node set, we impose a condition in the algorithm that a node attaches to a dominating node only if it is less than distance  $r(Y, X)$ . We note however that the above is not possible if  $r(Y, X)$  is greater than the actual communication range  $r$  (which also means that the network is not covered and hence we do not consider this condition)

Although this method gives us an active node set according to the required distribution condition, it cannot guarantee a connected set it is rendered useless for routing purposes and is meant only for comparison with the algorithm described in the next section.

### 5.2 Scheduling with Virtual Dominating Sets

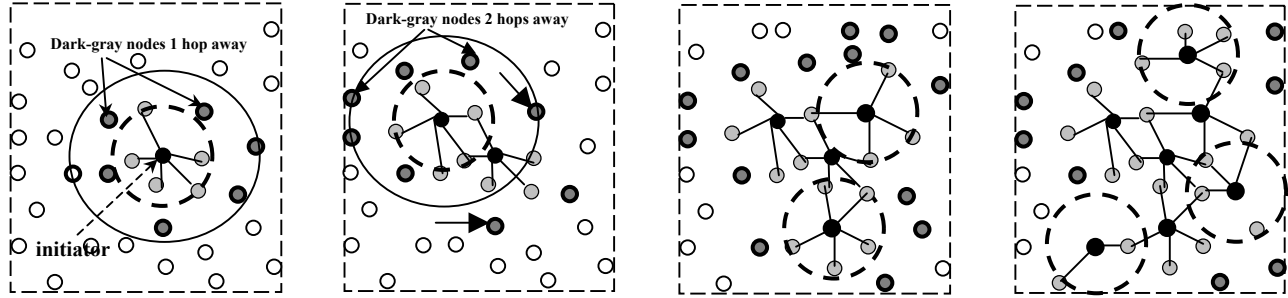
To construct a connected dominating set at the desired density we use the concept of *Virtual Dominating Sets (VDS)*. The concept of *VDS* was first introduced in [28] and was used for retrieving topology at multiple resolutions. In this paper we propose an algorithm for node scheduling closely related to the *VDS* concept and using their definitions on unit disc graphs.

Consider a graph  $G(V, E(r))$  where an edge exists between two nodes if their distance is at most  $r$ . For wireless networks, communication range,  $r$  defines the network graph. Consider the following definitions on  $G(V, E(r))$ :

**Virtual Edge Set ( $E(r_v)$ ):** The subset of  $E(r)$  such that each edge in the set has endpoints at most distance  $r_v$  apart. In this case,  $r_v$  is called the *Virtual Range*. The subgraph  $G(V, E(r_v))$  is referred to as the *virtual graph*.

**Minimal Virtual Connected Dominating Set  $MVCDS(r)$ :** An MCDS on graph  $G(V, E(r_v))$ .

Figure 11 illustrates the concept of *MVCDS*. In the first figure, we have the *virtual range* equal to the communication range ( $r=r_v$ ) and the *MVCDS* is given by the three black circles. In the second figure, the virtual range is much smaller than the communication range and many of the edges of the original graph are missing (shown by dotted lines). The *MVCDS* is given by the four black



**Figure 12 Illustration of the VCDS algorithm. The larger and the smaller circles depict the communication and the virtual range. The coloring takes place in layers (using timer mechanism). The dark gray nodes one hop away 1st change color and then forward a packet. Only after the 1st hop nodes change color, the 2nd hop nodes receive a packet and change color.**

circles in this case. We see that a decrease in *virtual range*  $r_v$  causes a decrease in the number of *virtual edges* and hence an increase in the cardinality of  $MVCDS(r_v)$ .

We note here that the *virtual range* is not the reduced transmission range but simply a definition on the graph, which allows for variable cardinality of dominating sets. The transmission range still remains  $r$  although the *dominating set* is created with range  $r_v$ . Intuitively, the virtual radius only acts as a control knob for the cardinality of the dominating set formed. Thus MVCDS is a relaxed optimization problem of the MCDS version and similar to *Weakly Connected Dominating Sets (WCDS)* [36]. However WCDS does not have a variable cardinality definition, which is important for node scheduling.

Next we describe the construction of the MVCDS. Any node randomly decides to initiate the process (the *initiator node*). Nodes to be kept active are selected in layers i.e. first the nodes one hop away from the *initiator* are selected and then nodes two hops away and so on. We will see that this layered scheme is required for good approximation bounds for the MVCDS. However it also creates a uniformly distributed set of active nodes. This is useful since it prevents any bottlenecks from forming (if the structure and the routing paths are too skewed) on the node scheduled graph.

1. The initiating node is the first node in the active node set and is colored black. The node then broadcasts a packet with its node id and color. All nodes which are within the virtual radius  $r_v$  of the black node are the *dominated nodes* and colored *gray* and all other nodes are colored *dark gray*.
2. A dark gray node changes its color to gray if it receives a message from a neighbor which has black color. Any node which becomes black, sends out a packet immediately to inform its neighbors which are colored according to step 1. If it does not receive any such message, it changes its color to black after a random delay based on its  $Id^3$
3. A black node formed in step 2 is at most one or two hops away from another black node from a previous step (currently the *initiator*), which it considers its *parent black node*. There is at least one intermediate node to connect any such parent-children pair. The intermediate node in our algorithm is the one from

which the current black node first received a packet from the previous hop. This intermediate node is also included in the active node set by including its ID when the black node forwards a packet.

4. Next all nodes (black, gray and dark gray) forward the request. Steps 1,2 and 3 are repeated until all nodes are colored gray or black. In the end we are left with a set of black nodes and some intermediate nodes connecting each black node to a black node formed in the previous step.

For a proper ordering of the layered coloring scheme (i.e., nodes with a lower hop distance from the *initiator* gets colored earlier) we use a forwarding delay for each node based on their hop distance from the *initiator*. Thus after the steps 1, 2 and 3, a node  $h$  hops away from the *initiator* forwards before a node  $h+1$  hops away. The coloring sequence is illustrated in Figure 12.

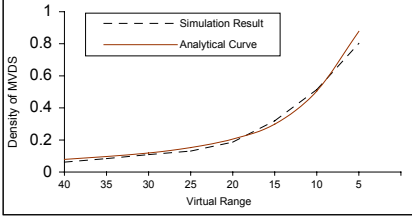
Based on the algorithm described we highlight the following properties of the VCDS. These are essential to derive the performance bounds for the algorithm

1. No two nodes change color at the same time since timer delays are unique based on unique node Ids. Hence, no two black nodes are virtual-neighbors of each other i.e. the black nodes actually form a Maximal Virtual Independent Set (MVIS)
2. The Virtual Connected Dominating Set (VCDS) consists of two sets of nodes, the MVIS (black nodes) and the connector node set. Each black node is connected to at least one black node through one connector node.
3. A node may have multiple neighboring active nodes (black or intermediate node) since the set may be created with a virtual range smaller than the real range. Thus at small  $r_v$  each node has a higher connectivity due to larger number of active nodes in its communication neighborhood.
4. Each node sends one messages after color changes and another after an intermediate node is selected.

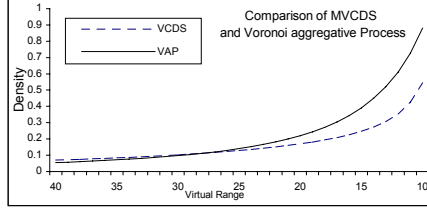
## 6. ANALYSIS OF THE VCDS

We first briefly look at the worst-case combinatorial approximation bounds of VCDS algorithm. Although the main aim of using VCDS instead of CDS is to have variable cardinality dominating sets, we note that the algorithm described can be used to create a normal CDS as well (by using virtual radius equal to communication radius).

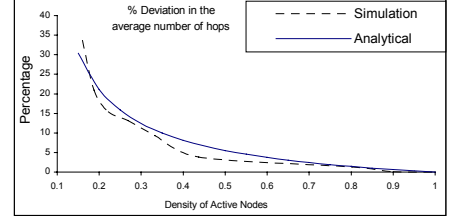
<sup>3</sup> E.g. the delay as a hash function on the node ID which will select a delay out of an  $O(n)$  time period. Note that this simple scheme would work without any time synchronization.



**Figure 13: Comparison of eqn. 22 and simulations. R=600m N=10000, r=40m.**



**Figure 14: Density of MVCDS with VAP, for R=600m N=10000, r=40m.**



**Figure 15: % increase in all pair shortest path lengths: R=600m, N=10<sup>4</sup>, r=40m.**

The VCDS algorithm achieves an approximation ratio of  $\delta opt-2$  with message complexity of  $O(n)$  and time complexity of  $O(n^2)$ . For details please refer to the Theorem A1, A2 and A3 in appendix. Let us compare our bounds with the two known algorithms for CDS with constant approximation bounds. In [30], the CDS construction has an approximation bound of  $192opt+48$  with message and time complexity  $O(n)$ . In [31], the CDS construction has approximation bound similar to ours ( $\delta opt$ ) but has a message complexity  $O(n \log(n))$ . To the best of our knowledge, our algorithm has the best approximation bound with linear message complexity.

Although the worst-case bounds are important for completeness of the analysis, from a practical standpoint it is not very useful since does not give any information about the expected cardinality and the node distribution of the VCDS formed (which are more important from the perspective of node scheduling at the required density). We now derive the results for the average cardinality of any VCDS formed using probabilistic methods.

We first look at the cardinality of the *MVIS* (the black node set) and then see how many intermediate nodes are added to form the *VCDS*. The black node set in the VCDS is a maximal virtual independent set. Thus a node in the graph can be black only if all its neighbors are *not* black. Consider different runs of the algorithm with different root nodes. Suppose a particular node  $i$ , becomes black,  $b_i$  times out of the  $k$  different runs of the algorithm. Then we define,

$p_i = \text{probability of node } i \text{ to become black for a randomly selected initiating node} = b_i / k \text{ for large } k.$

We assume that  $p_i$  is dependent only on the local degree of node ( $d_i$ ) and uncorrelated with the rest of the network. The independence assumption is not valid for a particular run of the algorithm but for a large-scale network and large number of runs. Thus the expected number of times each node becomes black across all runs of the algorithm is the expected number of times all its neighbors become gray. Then for each node the following set of equations are satisfied:

$$p_i = \prod_{j=1}^{d_i} (1 - p_j) \quad \text{where } p_j = j^{\text{th}} \text{ neighbor of } i. \quad \dots 17$$

Multiplying the set of equations for all  $i$ , each  $1-p_j$  appears  $d_i$  times in the set of equations. Thus we get:

$$\prod_{i=1}^N p_i = \prod_{j=1}^N (1 - p_j)^{d_j} \quad \dots 18$$

Let  $E[p_i] = \hat{p}$ . For large  $N$  and large number of runs  $k$ ,  $p_i$  for each node tends to the expected value  $\hat{p}$ . From law of large numbers, as  $\prod_{i=1}^N p_i \rightarrow \hat{p}^N$ , we get:

$$\hat{p}^N = (1 - \hat{p})^{\sum d_j} = (1 - \hat{p})^{Nd} \Rightarrow \hat{p} = (1 - \hat{p})^d \quad \dots 19$$

$d = Nr_v^2 / R^2 - 1$ , the expected virtual degree.

For sufficiently large  $d$  the above can be approximated using:

$$\hat{p}_{MVIS} = (\ln(d) - \ln(\ln(d))) / d \quad \text{for } d \geq 1 \quad \dots 20$$

Now we account for the extra intermediate nodes which connect the chosen set of nodes to form the connected tree. We see that any node is dominating with probability  $p$  (given by equation 19, 20) only if it falls outside the virtual range of another black node. Using area arguments, the probability that an intermediate node is also dominating is:

$$P = p_{MVIS} (r^2 - r_v^2) / r^2 \quad \dots 21$$

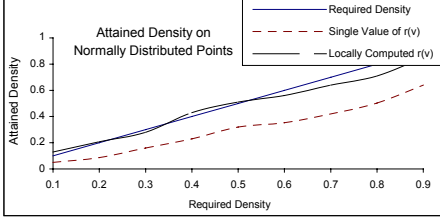
Hence the cardinality of MVCDS is given by:

$$|VCDS| = N p_{MVIS} (2 - p_{MVIS} (r^2 - r_v^2) / r^2) \quad \dots 22$$

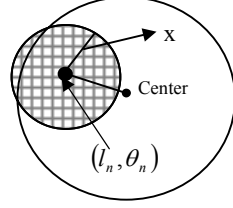
The expected cardinality of VCDS is thus  $O(N \ln(d) / d)$ , i.e., at large scales, it actually behaves like a *random  $d$ -regular graph* which also has a dominating set cardinality of  $O(N \ln(d) / d)$  [32].

Figure 13 shows the comparison between actual number of active nodes formed and the analytical expectation from equation 22. We see from the graphs that the uniformity assumption of the set of nodes in MVDS is fairly accurate. Next we compare the VCDS with VAP described in the previous section. Figure 14 compares the density of nodes formed using the Voronoi process and the MVCDS for different virtual ranges. We see that at small virtual ranges, the distribution deviates from the VAP. This is because of the connectivity constraints for MVCDS which leads to a large fraction of intermediate nodes with respect to the black nodes.

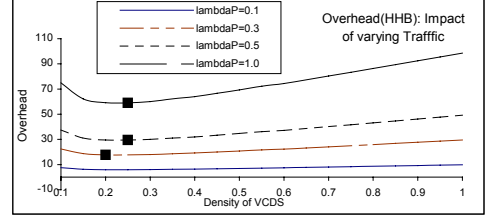
An important aspect of the derivations for optimal power consumption was the computation of the expected path lengths. The properties exhibited by VCDS should match those computed assuming a Poisson distributed topology in section 3. Figure 15 shows the percentage deviation in the number of hops from the optimal (when all nodes are active) with actual simulations and those computed using the formulation in section 3. We see that the simulation curves match the analytical results.



**Figure 16 Performance of the local virtual radius scheme with a Gaussian Distributed set of nodes.  $R=600m$   $N=10000$ ,  $r=40m$**



**Figure 17: Computing the distribution of distances for a node at position  $(l_n, \theta_n)$ .**



**Figure 18: Simulation Results: Overhead: Different traffic rates.  $N=10^4$ ,  $R=600m$ ,  $r=35m$ ,  $E_T=1$   $E_R=0.1$ ,  $PI=0.01$  and  $P_S=10^{-3}$  units/s,  $f=0.3$**

## 7. DISTRIBUTED COMPUTATION OF DENSITY CONDITIONS

Although the algorithm to create the VCDS is completely distributed, the density conditions were derived in a centralized fashion assuming uniformity in network parameters. In this section we show that the density conditions can be derived with only local information at each node. The distributed computation of the density condition is also important for adapting to local aberrations in network conditions which the centralized approach cannot take into account without global knowledge.

We first look at how the virtual radius should be computed locally to account for local aberrations in node density. Suppose we have  $N$  nodes with some arbitrary distribution and we want to reduce the number of nodes from  $N$  to  $NX$ , where the  $NX$  nodes are chosen according to some other distribution (based on optimal distribution). The use of a single virtual radius would not be sufficient due to non uniformity in the number of neighbors. We use the local degree instead of the average network degree to compute the virtual radius for each node. For a node  $n$  with local degree  $d_n$ , the local density is calculated as  $\delta_n = (d_n + 1) / \pi r^2$ . Let  $P_{MVIS}(n)$  denote the required distribution around the location of node  $n$ . The virtual radius for node  $n$  is computed as follows:

$$P_{MVIS}(n) = (1 - P_{MVIS}(n))^{\delta_n r_n(n)^2 - 1} \quad \dots 23$$

$$\Rightarrow r_v(n) = \sqrt{(\ln(P_{MVIS}(n)) / \ln(1 - P_{MVIS}(n)) + 1) / \delta_n}$$

Figure 16 shows the effect of using this scheme on a gaussian distributed set of 1000 nodes around the center of a  $400 \times 400$  m<sup>2</sup> field and variance 75m. The aim here was to create a node-scheduled topology with uniform density starting from the underlying gaussian density i.e. the required density distribution at each point in the network  $P_{MVIS}$  is constant. The curves show how the localized method adapts to the non-uniform distribution of nodes. Thus the local computation of the virtual graph can approximate and control the local density of nodes and thus create any arbitrary density on a large scale. Although we do not solve the node-scheduling problem for non-uniform distributions, the graph illustrates the potential of the local virtual degree method for handling non-uniform density.

Now we show how nodes can compute the optimal density conditions using only local knowledge of network parameters. Consider a node  $n$  with local degree  $d_n$ . We assume that the node has knowledge of the number of nodes in the network  $N$  and the total area of the network and hence can calculate the expected degree of any node as  $d$ . Thus  $n$  can compute the average hop length from equation 8. Next it uses the following formulation of the overhead equation to find the optimal density.

$$P_{HHR}^n(h) = \lambda_p O_{HHR}(h) (E_T + E_R \delta_n \pi r^2) d_n / d + P_I (1 - X_n) + P_S X_n \dots 24$$

The most important element in the above equation is the scaling of the traffic rate to  $\lambda_p d_n / d$ , using the ratio of local and the average network density. The global formulation assumes a uniform traffic rate originating in each region. But a region with higher density would have larger traffic rate which the local scaling actually attempts to take into account. It can be shown that as density increases this would in fact converge to the global computation. The node now computes the optimal density  $\delta_n^*$  from equation 25 and  $p_{MVIS}$  from equation 23, 24 and 26.

$$\delta_n^* = N p_{MVIS} (2 - p_{MVIS} (r^2 - r_v(n)^2) / r^2) / \pi R^2 \quad \dots 25$$

Finally the virtual degree  $r_v(n)$  for node  $n$  is computed as

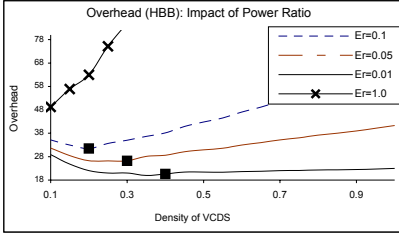
$$r_v^* = \sqrt{\pi r^2 (1 + \ln(p_{MVIS}) / \ln(1 - p_{MVIS})) / (d_n + 1)} \quad \dots 26$$

For non-uniform density distribution we have to separately consider the density distribution of distances of each nodes in the network. Let the position of  $n$  in the network be  $(l_n, \theta_n)$  as shown in Figure 17. Then the density of nodes which are at length  $x$  from node  $n$  is the intersection of the field circle with a circle of radius  $x$  centered at  $(l_n, \theta_n)$  as shown in the figure. We find the PDF of hop lengths from this node in exactly the same manner as in section 3 but use the local degree  $d_n$  to compute it since the node does not know the actual distribution in the network. Thus for expected hop length  $\hat{h}_n$ , for a node  $n$ , the local optimal density is computed by minimizing equation 24. Intuitively this does take care of non-uniform density as well but showing the convergence is beyond the scope of this paper.

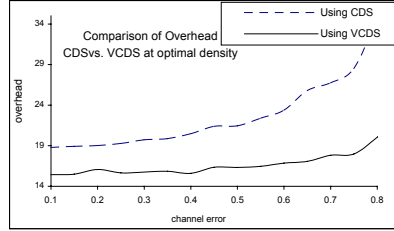
The modifications proposed in this section are extremely powerful for handling arbitrary conditions. In short, we have shown that it is indeed possible to achieve conditions of optimal power consumption in a distributed manner.

## 8. SIMULATION RESULTS

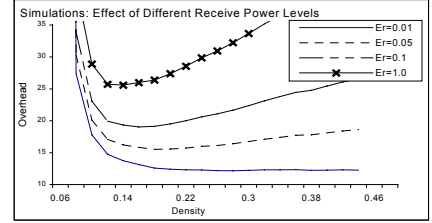
In this section we briefly describe simulation results to validate the theoretical analysis. We consider a network of 10000 nodes uniformly distributed in a circular field of radius 600m. The communication range is set to 35m. The simulations are based on the normalized and average power consumptions for all the nodes in the network. The values are averaged for 5 randomly generated topologies. We evaluate the power consumption of reliable packet delivery using HHR based on various parameters. Each node numerically computes the optimal density and the required virtual range to create the VCDS. For a normal CDS, we put communication range equal to virtual range and compare the overhead for each case. We first construct the VCDS at various



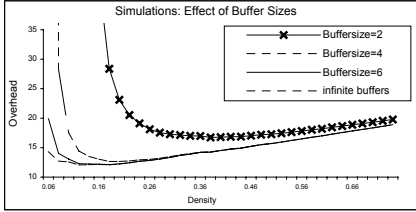
**Figure 19** Overhead on VCDS at different densities.  $R=600m$ ,  $r=30m$ ,



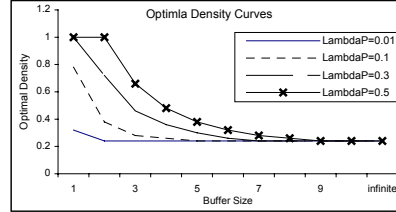
**Figure 20** The overhead at opt. density (VCDS) and min. density (CDS,  $r_v=r$ ).



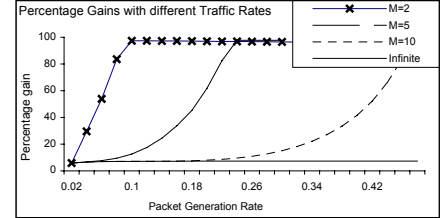
**Figure 21** Overhead for different Receive power levels.  $M=10$ ,  $\lambda_p=0.1$



**Figure 22** Simulations results with different buffer sizes at different densities.



**Figure 23** Optimal density curves for different traffic rates and buffer sizes.



**Figure 24** Percentage gains by using the VCDS vs. using CDS  $M=$  Buffer Size

Typical values of parameters:  $R=600m$ ,  $r=35m$ ,  $E_T=1J$ ,  $E_R=0.1J$ ,  $P_T=0.01$ ,  $P_S = 10^{-3} J/s$ ,  $\lambda_p=0.1$ ,  $f=0.1$

resolutions and forward packets reliably using the HHB scheme to find the overhead.

We first look at the overhead of HHB with unconstrained memory available in nodes. Figure 18 plots the overhead for different traffic rates. Figure 19 plots the overhead for different transmit and receive power levels. Depending on the network parameters the minimum overhead occurs at different densities. For example, when transmit and receive powers are equal, optimal power consumption occurs at low densities. On the other hand when the transmit power is large compared to the receive power it doesn't cost much to keep the receivers on. Thus scheduling nodes at the optimal density can save network power. Similar effect is illustrated in Figure 20 for different channel error rates, where we compare the performance of node scheduling at optimal density using VCDS vs. scheduling nodes using a normal CDS (with the network at the lowest density). We see that at optimal density, the overhead is always less than for the CDS case. We also see that with increasing channel errors, the low density performs badly since large hop lengths and reduced network degree significantly affects the overhead. The result can be put in perspective of other CDS construction algorithms as well since our algorithm has very good approximation bounds compared to other existing algorithms for constructing CDS.

Next we consider the problem with buffer constraints. Figure 21 plots the overhead of HHB for different buffer sizes and Figure 22 for receive power levels. Similar to our analytical curves, we see that the network becomes unstable at very low densities. The optimal density points vary widely depending on the network parameters. Figure 23 plots the optimal density points for different traffic rates. Figure 24 shows the percentage gains achieved by scheduling nodes at the optimal density when compared to the overhead at minimum density given by a normal CDS. The energy saving are really striking for high traffic or low buffer conditions.

The simulation results on a real topology verify analytical model of network power consumption under different network density.

The results also validate our claims that a minimal network density achieved with an MCDS is counter productive and that we can actually reduce the power consumption by orders of magnitude depending on the network conditions and parameters by keeping a high percentage of nodes active.

## 9. RELATED WORK

We first give a brief literature survey related to transmission power control. In [1] and [2] the effect of transmission radius on the average path lengths in a multi hop wireless network was first studied and the range for minimum power consumption was derived. PAMAS [3] then proposed the idea that a node may choose a smaller number of hops using a longer range so as to reduce power. The concept was extended in [4], to route along minimum energy paths using different communication ranges and find the optimal paths for different reliability schemes. Transmission power control for TCP flows in MANETs was studied in [8] and [9].

Scheduling the minimum set of nodes to be active for routing purposes is usually done with an approximation of Minimum Connected Dominating Set (MCDS). Routing over MCDS is usually referred to as Spine Routing or Backbone routing and [10], [11], [12], [13], [14], [15] and [16] are some well known algorithms using this concept. CDS reduces the routing overhead and minimizes the idle power dissipation in nodes but may result in degradation of the network quality (in terms of capacity, delay, overhead, coverage etc.). Reference [17] looks at the problem of load sharing and energy dissipation in nodes to reduce unequal depletion of energy in cluster heads of the CDS. Reference [18] tries to conserve the network coverage in sensor networks while minimizing the number of active nodes. Reference [19] proposes a localized algorithm called SPAN for node scheduling. Using simulations they show that the node-scheduled topology does not suffer too much in terms of latency and capacity of the network while reducing redundant power consumption. However none of the papers consider the impact of node scheduling on the overhead of reliable transmissions for ad hoc networks.

The analysis for average path lengths is similar to ones in [1] and [2] with some important differences. In our analysis we consider a radial distance of progress towards the sink (which means all forms of trajectories between source and sink are considered in the probability distribution) whereas in [1] and [2] the progress is assumed along a straight line joining the source and sink. Also we consider the actual probability distribution of distances in bounded regions instead of the expected distance. Thus we get a more accurate estimate of the average hops.

To the best of our knowledge, our MCDS approximation algorithm for node scheduling ( $8opt-2, n$ ) is the only algorithm with constant approximation bounds along with ones in described in [30] ( $8opt-2, O(n \log n)$ ) and [31] ( $192opt-48, O(n)$ ). A parameterized approximation algorithm to create CDS appears in [33]. However the parameter is used to control the optimization bound and is not useful for controlling the network density. A concept similar to the virtual radius is used in [35] to form clusters given a specified radius of coverage by cluster heads. The algorithm has an approximation bound of  $O(n^{1/2})$  with message complexity of  $O(n^2)$  for a two dimensional network. However, both [33] and [35] do not provide any density distribution results which might have proved useful while applying the methods for node scheduling. Finally, the concept of MVCDS is comparable to the concept of *Weakly Connected Dominating Sets* (WCDS) [36]. WCDS is a relaxed formulation of the MCDS problem creating a larger cardinality CDS than a normal MCDS. Compared to all the above, our algorithm is ideal for node scheduling since it has the best approximation bounds along with the ability to accurately control the network density.

## 10. CONCLUSIONS AND FUTURE WORK

In topology control based on node scheduling, only a minimum number of nodes needed for routing are kept active. We show that such a formulation, often realized using an MCDS is sub optimal for packets transmissions. This is because when a large fraction of nodes are inactive for routing purposes, the actual path lengths increase dramatically. With such an increase in path lengths, the impact on overhead for reliable delivery is even more severe. Algorithms such as HHB, which utilize the local node density, are the worst affected by low density of scheduled nodes. Finally, with limited buffers, minimal number of nodes may result in network instability. Our results show that being too aggressive in switching off redundant nodes is counter productive.

Instead, we propose a simple node-scheduling scheme based on the novel concept of minimum virtual connected dominating sets (MVCDS). This scheme schedules nodes at the optimal density as opposed to the minimal density. We show that MVCDS provides an active node set with connectivity guarantees while closely approximating the optimal density conditions. The algorithm is fully distributed and message optimal and is also shown to be adaptive.

Finally, we believe the results in the paper provide guidelines for future research on topology control using the node-scheduling scheme both in terms of the theoretical analysis as well as practical implementation. To limit the scope of the work certain simplistic assumptions were considered. For example we assumed that transmission ranges were fixed, and channel errors, network traffic and node distributions were uniform across the network. By relaxing these assumptions, an interesting and perhaps a non-trivial study would be to find optimal conditions for power

consumption when both node scheduling and transmission power control is available. This is a subject of future work.

## 11. ACKNOWLEDGEMENTS

This work was supported in part by NSF grant ANI-0240383.

## 12. REFERENCES

- [1] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks, or why Six is a Magic Number." IEEE National Telecommunications Conference}, 1978.
- [2] Takagi, H. and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals", IEEE Trans. on Communications, Vol. COM-32, No. 3, pp. 246-257, March 1984.
- [3] S. Singh and C. Raghavendra, "PAMAS: Power Aware Multi-Access protocol with Signaling for Ad Hoc Networks", ACM C.C.R., 1999."
- [4] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks", In Proc. of the ACM MobiHoc, 2002.
- [5] S-J. Park, R. Sivakumar, "Quantitative Analysis of Transmission Power Control in Wireless Ad-hoc Networks", 2002 International Conference on Parallel Processing Workshops (ICPPW'02)
- [6] S. Agarwal, S. V. Krishnamurthy, R. H. Katz, S. K. Dao, "Distributed Power Control in Ad-hoc Wireless Networks" Personal Indoor Mobile Radio Conference Oct. 2001
- [7] Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky, "NetworkLifetime and Power Assignment in Ad-Hoc Wireless Networks", in ESA 2003.
- [8] I. Ali, S. Bansal, R.Gupta, A. Misra, R. Shorey and A. Razdan, "Energy Efficiency and Throughput for TCP Traffic in Multi-Hop Wireless Networks", Proceedings of IEEE INFOCOM 2002, June 2002, New York City, USA.
- [9] L. Li and P. Sinha, "Throughput and Energy Efficiency in Topology-Controlled Multihop Wireless Networks", Proc. ACM (WSNA-MOBICOM), September, 2003.
- [10] B. Das, V Bhargavan, "Routing in Ad Hoc Networks Using Minimum connected dominating Sets". IEEE International Conference on Communications (ICC '97), June, 1997.
- [11] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A design concept for triable mobile radio networks with frequency hopping signaling." Proc. of IEEE, 75, 1 (01/87) 56-73.
- [12] M. Gerla, J.T. C. Tsai, "Multicluster, mobile, multimedia radio networks," ACM J. Wireless Networks, 1, 3 (1995) 255-265.
- [13] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," European Symposium on Algorithms. 179-193, 1996.
- [14] P. Sinha, R. Sivakumar and V. Bhargavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm" In proceedings of IEEE Infocom 1999.
- [15] V. Rodoplu and T.H.Meng, "Minimum Energy Mobile Wireless Networks", IEEE JSAC, Vol. 17, August 1999.
- [16] S. Bannerjee, S. Khuller, "A Clustering Scheme for Hierarchical Control in Wireless Networks", In Proceedings of IEEE INFOCOM 2001.

- [17] J. Wu, F. Dai, M. Gao, I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", *Journal of Communications and Networks*, Vol. 4, No.1, March 2002
- [18] D. Tian, N. D. Georganas, "Network-Provisioning: A coverage-preserving node scheduling scheme for large wireless sensor networks" *Proc. ACM (WSNA-MOBICOM)*, September, 2003.
- [19] B. Chen and K. Jamieson and H. Balakrishnan and Robert Morris, "SPAN: An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks", *Mobile Computing and Networking*, pages 85-96, 2001
- [20] Y. Xu, J. Heidemann, D. Estrin. "Geography-informed Energy Conservation for Ad Hoc Routing". In *Proceeding of MOBICOM'01, Italy 2001*; 70-84
- [21] S. Bandhopadhyay, and E. J. Coyle, "An Energy Efficient Hierarchical Clustering algorithm for Wireless Sensor Networks", *IEEE INFOCOM 2003*.
- [22] S. G. Foss and S. A. Zuyev, "On Voronoi Aggregative Process Related to a Bivariate Poisson Process", *Advances in Applied Probability*, Vol. 28 no. 4, pp 965-981, 1996
- [23] Marathe, M. V., Breu, H., Hunt III, H. B., Ravi, S. S., and Rosenkrantz (1995), D. J. (1995), "Simple heuristics for unit disk graphs", *Networks* 25, 59-68.
- [24] I. Kang and R. Poovendran, "A Novel Power-Efficient Broadcast Routing Algorithm Exploiting Broadcast Efficiency with Omnidirectional and Directional Antenna", *IEEE -VTC*, Oct 6-9, 2003, Orlando, FL
- [25] A. Srinivas and E. Modiano, "Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks", *MobiCom 2003*
- [26] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks," *Mobile Networks and Applications (MONET)*, 6-3, pp. 251-263, June 2001.
- [27] B. Deb, S. Bhatnagar and B. Nath, "Information Assurance in Sensor Networks" in *Proceedings of WSNA-03, MOBICOM-03*, San Diego 2003
- [28] B. Deb, S. Bhatnagar, B. Nath, "Multi-Resolution State Retrieval Sensor Networks", *1st IEEE Int. Workshop on Sensor Network Protocols and Applications* May 11, 2003
- [29] K. Arisha, M. Youssef and M. Younis, "Energy-aware TDMA-based MAC for Sensor Networks", In *Proc. of IEEE IMPACCT*, 2002
- [30] K. M. Alzoubi, P.-J. Wan, O. Frieder, "Message-Optimal Connected-Dominating-Set Construction for Routing in Mobile Ad Hoc Networks", *MobiHoc 2002*
- [31] P.J. Wan, K. M. Alzoubi, O. Frieder, "Distributed Construction of Connected Dominating Sets in ad hoc wireless networks", *IEEE INFOCOM 2002*
- [32] N. Alon and J. Spencer, "The Probabilistic Method", *John Wiley* as part of the *Inter-science Series in Discrete Mathematics and Optimization*, 2000
- [33] F. Kuhn and R. Wattenhofer. Constant-time distributed dominating set approximation. In *Proceedings of the 22nd*

ACM Symposium on Principles of Distributed Computing (PODC'03), July 2003.

- [34] T. Moscibroda and R. Wattenhofer, "Maximizing the Lifetime of DominatingSets", *IEEE- WMAN 2005*.
- [35] Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. In *Proc. 17th ACM Symp. on Computational Geometry*, Jun 2001.
- [36] J. W. Grossman, "Dominating Sets whose Closed Stars form Spanning Trees", *Discrete Mathematics*, 169(1-3) (1997)
- [37] P. J. Shwartz, S.S. Lam., "Buffer overflow in a store and forward network node", *IBM J. Res. Dev.* 1976

### 13. APPENDIX

We compute the worst-case bounds for the VCDS algorithm. If the virtual graph a connected sub-graph of the original graph., let  $opt = |MVCDs|$  be the cardinality of the minimum virtual connected dominating set.

**LEMMA A1:** Any MVIS on  $G(V, E(r_v))$  is at most  $4opt+1$ .

**PROOF:** The case for any MIS on a unit disk graph was proved in [31]. The case for virtual graph is not different since the MIS is created on a disk graph of virtual radius and hence the result holds for the virtual graphs as well.

**THEOREM A2:** If the virtual graph is connected, the size of the Virtual connected dominating set (VCDS) is within a constant approximation bound given by  $|VCDS| \leq 8opt - 2$ .

**PROOF:** Let  $|MVIS|$  be the cardinality of the maximal independent set of black nodes. The black node tree would consist of at most  $|MVIS|-1$  intermediate nodes. Hence the cardinality of VCDS is at most  $2|MVIS|-1$ . From lemma 1, and that there is always one initiating node, the approximation bound for the VCDS is given by  $|VCDS| \leq 8opt - 2$ .

**THEOREM A3:** The VCDS algorithm has a message complexity of  $O(n)$  and a time complexity of  $O(n^2)$ .

**PROOF:** From the algorithm description we see that each node whether a black node or a gray node sends *at most two* messages. Thus the message complexity is  $O(n)$ . For the time complexity, we note that each node needs to have a unique time delay out of an  $O(n)$  time period to change color since otherwise two or more nodes can become black at the same time. Since the maximum number of hops is  $n-1$ , the worst case time complexity is  $O(n^2)$ . However the time complexity can be reduced to  $O(nd)$  by considering the degree of each node where  $d$  is the maximum degree of the graph. We leave the discussion due to lack of space.

The CDS approximation algorithm described in [31] relies on a spanning tree structure to attain the optimality bound. In [31], the spanning tree construction itself takes  $O(n \log(n))$  steps. In our case the spanning tree is implicitly created while the algorithm progresses. This is achieved with only  $n$  messages using unique delays for each node and layered forwarding based on hop lengths from the initiator. This results in the worst time complexity of the three algorithms but the best message complexity. We note however that this is only possible in broadcast networks where only one message is required to communicate with all neighbors. We also note that the implicit spanning tree is useful only to achieve good approximation bounds for the algorithm and need not be maintained or used for routing.