Using Adaptive Range Control to Maximize 1-Hop Broadcast Coverage in Dense Wireless Networks

Xiaoyan Li, Thu D. Nguyen, Richard P. Martin
Department of Computer Science, Rutgers University
110 Frelinghuysen Rd, Piscataway, NJ 08854
Email: {xili,tdnguyen,rmartin}@cs.rutgers.edu

***Abstract*— We present a distributed algorithm for maximizing 1-hop broadcast coverage in dense wireless sensor networks. Our strategy is built upon an analytic model that predicts the optimal range for maximizing 1-hop broadcast coverage given information like network density and node sending rate. The algorithm allows each node to set the maximizing radio range using only the locally observed sending rate and node density. The algorithm is thus critically dependent on the empirical determination of these parameters. Our algorithm can observe the parameters using only message eavesdropping and thus does not require extra protocol messages. Using simulation, we show that in spite of many simplifications in the model and incomplete density information in a live network, our algorithm converges fairly quickly and provides good coverage for both uniform and non-uniform networks across a wide range of conditions. We also demonstrate the utility of our algorithm for higher layer protocols by showing that it significantly improves the reception rate for a flooding application as well as the performance of a localization protocol.**

I. Introduction

We consider the problem of how networks of extremely dense sensors connected by a wireless network can adjust their output power to maximize the number of 1-hop receivers of a broadcast message. Many recent sensor network protocols, such as directed diffusion [1] and ad-hoc positioning [2], rely on periodic broadcasts. In addition, many protocols, such as code propagation [3] and dynamic source routing (DSR) [4] rely on aperiodic broadcasts. Indeed, the fundamental broadcast nature of wireless networks makes broadcast an ideal building block for the discovery, routing and localization functions, which will be critical for future sensor network systems.

At the same time, as technology trends continue to reduce the size, power consumption and cost of embedded wireless devices, the scale of the networks built with them will continue to grow. For example, sensors are rapidly approaching the small size and low cost necessary for novel uses such as *active tagging*; i.e., representing the state of common everyday objects such as furniture, pens, and books. Our initial measurements of a common indoor environment for the active tagging application shows that the average degree of such a network could easily range up to several hundred.

Given the high densities of these future sensor networks, a resulting challenge for applications using broadcast will be how to manage channel capacity to ensure good performance in terms of throughput, fairness and broadcast coverage. This challenge arises because if all nodes act greedily, using the maximum range, the channel will collapse; that is, the likelihood of any neighbors receiving the message correctly quickly approaches zero in dense networks due to collisions. As shown in [3], many protocols perform poorly as density increases because of this issue.

We are thus motivated to consider how devices in these networks can maximize the number of 1-hop receivers of a broadcast message. We refer to *broadcast* as a 1-hop message, as opposed to *flooding* which intends to cover all the nodes via multi-hop forwarding. We choose to examine broadcast coverage because it is an important metric for protocols using broadcast as a building block. In addition, a network without good broadcast coverage is unlikely to have good throughput or latency properties — if many nodes fail to receive a broadcast message they would most likely have trouble receiving a normal unicast message due to the fundamental channel properties (i.e., a single sender impacts many receivers) of wireless transmission. Although our scheme may not result in optimal performance when applied to multi-hop applications, e.g. flooding, we expect it still benefits many simple multi-hop protocols because it improves channel utilization and prevents collapse. We show in Section IV-B.1 that applying our range control algorithm significantly improves the performance of a naive flooding scheme without introducing much additional complexity or any multi-hop information.

Our approach centers on the *spatial reuse* of wireless resources. Specifically, each node considers the surrounding sending rate, node density, and a simple geometric model of wireless communication to compute the radio range, which probabilistically maximizes the *coverage* for a broadcast packet. Informally, coverage is the number of nodes that correctly receive a 1-hop broadcast packet. Our algorithm uses radio range as a parameter because it is more tractable to analyze than output power directly; adjusting output power to achieve the computed optimal range remains future work.

The work in [5] presented an strict analytic model that predicts the optimal range for maximizing 1-hop broadcast coverage given information like network density and sending rate. The contribution we present in this work is a distributed algorithm that uses the model in order to arrive at the optimal range. The challenges in turning the model into a workable algorithm arise from a combination of modeling simplifications and the use of parameters which are difficult observe. The

model assumes there is no convergence period, and that both node location and packet arrivals follow a Poisson distribution. The model also assumes the nodes have perfect information on the spatial density. We show that our algorithm is robust because even when the assumptions do not hold and the nodes do not have exact density information, the algorithm converges quickly and maintains statistically good coverage.

Our range control approach is quite applicable to small wireless devices because of its simplicity; a device only needs to observe the sending rate and spatial density to use the algorithm. Also, the analytic model moves the complexity offline in a numeric solver leaving the nodes to only perform a lighter weight extrapolation.

To validate our approach we constructed a large-scale, three-dimensional simulation environment. We simulated our algorithm on both a uniform density environment, as well as on a more realistic environment based on location measurements of real objects in a laboratory in our department. We found that our approach is quite effective at controlling nodes ranges under a wide range of conditions. Our algorithm converges quickly, is stable, and delivers statistically fair coverage for many starting conditions as well as for both uniform and highly non-uniform topologies. To show the potential benefit of range control to higher-level services, we study the impact of our algorithm on a flooding and a hop-by-hop localization protocol. We show that in both cases, range control significantly improved the efficiency of the higher-level service.

Our contributions are summarized as follows:

- A range control algorithm that only uses local observations to estimate density and sending rate.
- A demonstration of the algorithm's robustness to initial conditions and non-uniform densities.
- A demonstration of the impact of range control on two higher layer services.

The remainder sections of this paper are organized as follows: In Section II, we summarize the analytic model. Section III is a description of the algorithm and discusses how we overcome real-world difficulties observing the inputs needed by the analytic model. In Section IV we evaluate the algorithm with respect to convergence, robustness, and coverage properties. We compare our work to related work in Section V. Finally, in Section VI we conclude.

II. Analytic Model

In this section we summarize the analytic model presented in [5]. Recall the model derives an optimum transmission range, R_o, which maximizes the 1-hop broadcast coverage, based on the sensors' sending rate and density. We first clarify the assumptions of the model. Because the model can only be solved numerically, we then introduce an extrapolation rule which the model uses to compute the range given an initial pre-computed optimum.

A. Model Assumptions

The model uses a set of assumptions to make a solution tractable to solve: (1) nodes are spatially distributed according to a Poisson distribution with an average density of λ_s; (2) applications running on each node generate packets to send according to a Poisson distribution with average rate λ_p; (3) all packets are of the same length and take time T to transmit; and, (4) all sensor nodes use a CSMA protocol. It also uses a fairly simple wireless communication model similar to the one in [6]: all nodes have the same radio range R, where nodes within R distance from a transmitter will detect the packet transmission while those further away will not. More than one packet transmission within distance R to a receiver will cause collision and all overlapped packets at the receiver are corrupted.

B. Extrapolation Rule

Given the assumptions in Section II-A, the authors in [5] first built a geometric model that predicts the likelihood of collision given all nodes in a uniform density network broadcast at the same rate and range. The resulting model can be numerically solved to find the optimal range for any combination of density and broadcast rate. Unfortunately, the model can not be solved analytically because the range is not computable using a closed-form formula. Thus, they next introduce two relations from which an extrapolation can be used in real-time to predict the optimum range for any rate and density given a single precomputed optimum. The extrapolation rule (in 2 dimensional scenarios) is:

$$R_o = \sqrt{\frac{C_o}{\lambda_s \left(1 - e^{-\lambda_p 2T}\right)}}$$

where C_o is computed as $\lambda_s^{'} \left(1 - e^{-\lambda_p^{'} 2T}\right) R_o^{'2}$ for the pre-computed optimum $R_o^{'}$ for scenario $(\lambda_s^{'}, \lambda_p^{'})$.

Validation through simulation demonstrated that the extrapolation predicts R_o to within 16%. We show in this work that this level of accuracy is sufficient to apply this model in practical settings.

III. Implementation

In this section we present our distributed algorithm for realizing dynamic range control for broadcast messages in dense wireless networks. Our approach basically operates by setting the range to R_o according to the analytic model based on observed sending rate and density information. It thus stands in contrast to feedback-control style algorithms, which would adjust R as response to the observed coverage. Critically, our approach depends on accurate estimation of density and sending rate. We first provide an overall description of the algorithm and then present the details of the techniques we used to estimate the local density.

A. Overview

Figure 1 shows the pseudo-code of our algorithm. It follows a simple pattern of adjusting the output power to set the range (R) every *adjustment interval*, which we currently have set at 20 broadcasts. Currently, the sending rate used by the algorithm is estimated by observing the packet arrival

```
Procedure RECEIVE_BROADCAST
  IF (push or pull message)
        add sender as a push or pull neighbor
  ELSE
        add sender as a normal neighbor
                        ...
Procedure SEND_BROADCAST
                        ...
  IF (packets_sent_since_last_adjustment == 20)
        call ADJUST_RANGE()
        packets_sent_since_last_adjustment = 0
                        ...
  IF(this is a push message slot)
        packet_range = min(R_max, distance to the
            furthest neighbor excluding neighbors
            resulted from received push messages)
  ELSE IF(this is a pull message slot)
        packet_range = min(√2R,R_max)
  ELSE
        packet_range = R
                        ...
  packets_sent_since_last_adjustment++
                        ...
```

```
Procedure ADJUST_RANGE
1. Remove expired neighbor information
2. λ_p = Total_Packets_Sent/Total_time
  E(N) = π C_o/(1−e^(−λ_p 2T))
  IF(number of neighbors < E(N))
        IF(within last interval:
            no correct packets reception AND
            observed transmission on the channel)
                R_next = R/√2
        ELSE
                R_next = min(√2R,R_max)
  ELSE
        filtered_neighbors =
            all_neighbors excluding neighbors
            from push/pull that are outside of √2R
        IF(number of filtered_neighbors < E(N))
            useful_neighbors = all_neighbors
        ELSE
            useful_neighbors = filtered_neighbors
        λ_s = median density from useful_neighbors
        R_next = √(C_o/(λ_s*(1−e^(−λ_p 2T))))
3. Schedule the next push/pull message if needed
```

Fig. 1. *Range control algorithm. The algorithm works by snooping on each broadcast packet transmission and reception. Adjusting the current range, R, is performed every 20th packet transmission.*

rate from the local node's application over the interval; this assumption is valid if all nodes are running a similar set of applications. We leave as future work to see if using the observed input rate is accurate and fair enough when nodes have different broadcast rates. If local information proves insufficient, combining observed channel utilization with neighbor information to infer average sending rate could also be used.

Node density is inferred based on neighbor information observed through packet eavesdropping. We found that observing density with sufficient precision and minimal overhead in this distributed setting is quite a challenge for a number of reasons. First, if R results in either a too sparse or too dense network, the number of observed neighbors will be too small to make a correct density estimation. In the first case, not enough neighbors will be observed. In the other case, too many collisions will cause many packets to be corrupted. In spite of the similar low number of observed neighbors, the correct adjustment action is opposite in the two cases. Second, outliers and skewed distance distributions adversely distort the density computation if not accounted for. Third, whether or not a node can observe a neighbor depends not on its range, but rather depends on its neighbor's range. In the remainder of this section, we describe how we use warm-up periods, collision monitoring, long-range messages and outlier consideration to overcome the challenges to correct density estimation. (Again all the discussions refer to the 2 dimensional scenarios.) We show in Section IV that the algorithm illustrated in Figure 1 is able to achieve accurate enough parameter estimation and the only network overhead it introduces is to extend the range of some existing broadcasts.

B. Density Estimation

In order to compute the density, the range control layer eavesdrops all packets and builds a list of all neighbors, as is shown in the RECEIVE_BROADCAST procedure in Figure 1. We assume the distance to each neighbor can be derived with a signal strength [7] or timing-differential signal approach [8]. Currently the density estimation uses only the distance information; clearly it would be even more accurate if the exact positions were known. In order to accommodate changing conditions, neighbors in the neighbor list are treated as soft state and expire if not refreshed by an incoming broadcast within 5 adjustment intervals (100 broadcasts). However, if a neighbor was added due to a push or pull message (as shown in *Procedure RECEIVE_BROADCAST*), it expires in 25 intervals.

a) Warm-up periods.: A node estimates density as $\lambda_s = \textit{number of neighbours}/area$. Thus a node needs to ensure it has sufficient neighbors for an accurate estimate. Based on observed sending rate, a node can calculate the expected number of neighbors covered in an ideal setting according to the formula ($E(N) = \lambda_s \pi R_o^2 = \pi \frac{C_o}{(1-e^{-\lambda_p 2T})}$). We use this number as a threshold that roughly indicates having heard from a sufficient number of neighbors. A node only starts adjusting its range according to the extrapolation model when its neighbor list reaches this number.

b) Collision monitoring.: If size of the list is below the threshold set for the warm-up periods, the node assumes the network is either too sparse or too dense. If within the last interval it has not received any correct packets, but did observe some channel utilization, the network is considered too dense using the current R. The node decreases R to cover half of the area it covered the previous adjustment interval. Otherwise, it

increases the range to double the area to cover more nodes. Clearly this is only a heuristic rule in case of ambiguity. Except for the extreme case, where collisions corrupt all packet receptions, it mostly favors an increase in the range. In Section IV we will show that for most cases it is effective in adjusting node ranges quickly to obtain an accurate density estimate. The adjustment for the worst scenario with regard to this heuristic (where nodes start with extremely large range for the network setting) did take a bit longer, but it eventually converged.

Although increasing a node's range does not by itself improve its number of observed neighbors, we will see that by employing long-range messages, increasing a given node's range causes neighboring nodes to increase their ranges for long-range messages as well. This implicit exchange caused by increasing a node's range thus ultimately increases the number of neighbors observed.

c) Occasional long-range messages.: Most of the broadcasts for a node use its local range setting (R). However, periodically the node needs to increase the range for some packets so that neighbors can correctly estimate density. We call these extended ranged broadcasts *push* and *pull* messages.

A *pull-ranged message* is used to account for a node's hidden terminals. A good range control algorithm should consider not only the immediately affected nodes, but also the hidden terminals. A pull message range is set to $\sqrt{2}$ times current range ($\sqrt{2}R$), which effectively solicits information from all the neighbors within that range. Although the hidden terminal effect extends to 2R, we found that this range creates more overhead than necessary for accurate density estimation. Recall that for uniform densities, doubling the range increases the number of affected nodes by a factor of 4. We found that setting the range of a pull message to $\sqrt{2}R$, thus doubling the area, is a good compromise between overhead of channel utilization and estimation of the effect of hidden terminals.

A *push-ranged message* is used to counteract the unidirectional information flow resulted from non-uniform ranges. The rational is that during range control a node should consider all the nodes its transmission affects, even if their ranges are too short for it to normally receive their messages. A push message range is set to the distance to the furthest neighbor, computed as the longest range received from either regular broadcasts or pull messages — but not push messages. Notice push message also addresses the information solicitation from pull message, which is necessary to realize pull message functionality. A situation where push ranges are needed is when a transmitting node, X, is in a sparse region and its regular messages extend into a cluster. A node in the cluster, Y, will thus set its push range to reach node X, ensuring that X will account for node Y in its density estimation. Notice that the small range of Y means its pull messages may not reach X.

We found that 1 push and 1 pull ranged broadcast out of 100 normal ones (1 in 5 intervals) is sufficient. Although only 2% of the broadcasts have these longer ranges, to avoid channel overload due to synchronization effects we randomize the timing of these longer broadcasts. Each node picks a random slot in an interval every 5 intervals and the starting interval is picked randomly.

d) Outlier consideration.: Because of non-uniform ranges, we found incorrect density estimation is likely if a node uses neighbors that are either too close or too far to estimate the density. That is, quite non-uniform distance distributions are possible. In order to prevent skew impacting the estimation, when we have enough neighbor information (as defined by warm periods threshold), neighbors received from push and pull ranged broadcasts are not added unless their distance is $< \sqrt{2}R$. (In order to differentiate these neighbors, we need to recognize push and pull messages from normal broadcasts; thus our algorithm requires an additional 2 bits in the message header.) In addition, we truncate the density estimate at the node with the median density. That is, we compute the density starting with the closest node, and get successive density estimates by including nodes one at a time. We then take the median density result as the final density. Simulation results show that the median is fairly effective in dealing with skewed distance measures.

IV. Experimental Evaluation

In this section we evaluate our range control algorithm in 3 dimensions via simulation. Our simulator models 3 dimensional spaces. The network configuration and wireless communication follow the same basic model as described in Section II-A. To account for boundary effects, we implemented spatial wrap-around, where transmissions propagate at the edge of the space as if space is a torus. In all of the experiments, we make sure that the transmission range is fairly small compared to the modeled space so that the spatial correlations resulted from wrap-around do not significantly affect the results.

We first focus on our algorithm's robustness. We show the algorithm converges fairly quickly and is stable across a large span of initial range settings and sending rates. We also show our algorithm is robust to highly non-uniform networks, quickly converging and allowing all nodes to achieve good coverage.

In the second half of this section, we demonstrate that range control can significantly impact the efficiency of higher-layer protocols/services. In particular, we study the impact of range control on two higher-layer services: an unreliable flooding and a hop-by-hop localization service. We demonstrate that both of these services benefit significantly from range control.

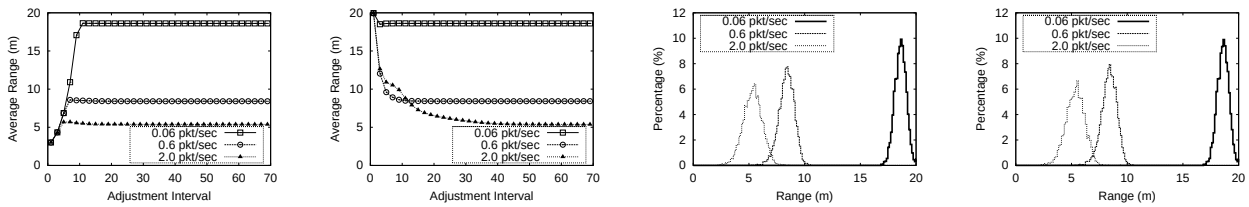
A. Algorithm Robustness

An important property of a distributed range control algorithm is its robustness to wide differences in range settings. In our context, we define robust to mean that nodes converge quickly to the ideal ranges and remain there independent of their starting states. We first run a set of simulations in a uniform network to show that the algorithm is fairly robust when the initial range settings are either too long or too

λ_s	λ_p	Analytic R_o	$D(R_o)$	R_{start}	D (R_{start})	$D(R_{start}) : D(R_o)$ in magnitude	Experiment number	Result: Average Range	Result: Error to R_o
0.006	0.06	18.70	163.46	3	< 1	$1 : 100$	1	18.61	0.5 %
				20	200	$1 : 1$	2	18.61	0.5 %
	0.6	8.74	15.8	3	< 1	$1 : 10$	3	8.41	3.8 %
				20	200	$10 : 1$	4	8.42	3.7 %
	2.0	5.96	4.33	3	< 1	$1 : 1$	5	5.38	9.7 %
				20	200	$100 : 1$	6	5.38	9.7 %

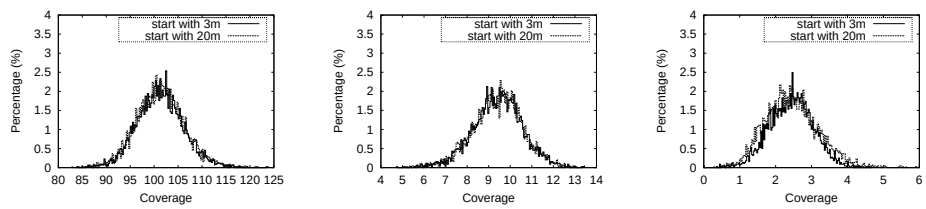
TABLE I

Experiment Settings for Robustness Evaluation. The unit for λ_s is $nodes/m^3$, for λ_p is $pkts/sec$ and for R is m. D(R) represents Expected Degree for nodes with range R.



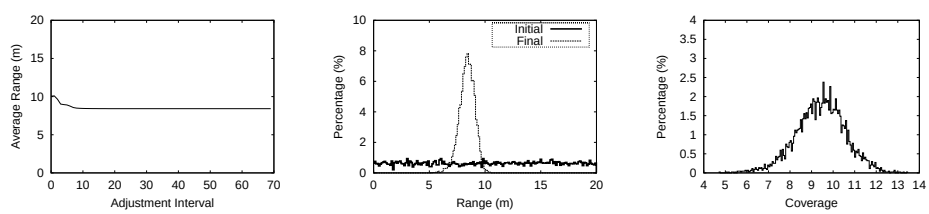
(a) Initial R=3m (b) Initial R=20m (c) Initial R=3m (d) Initial R=20m

Fig. 2. *Range adjustment for uniform density network.*



(a) λ_p = 0.06 pkt/sec (b) λ_p = 0.6 pkt/sec (c) λ_p = 2.0 pkt/sec

Fig. 3. *Histograms of coverage for uniform density networks.*



(a) Average Range (b) Range Histogram (c) Coverage Histogram

Fig. 4. *Random initial ranges. λ_p = 0.6 pkt/sec.*

short. Next, we show the algorithm is robust to random range settings.

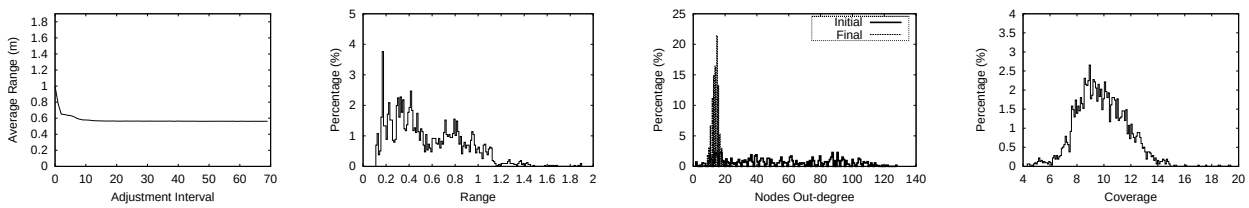
In addition to the initial state of the transmission range, a second condition we examine with respect to robustness is the impact of non-uniform density. Recall that the analytic model derived the optimal range setting assumes a uniform density; we do not expect future ad-hoc networks, especially those with non-stationary tagged objects, to be uniform. We found that our algorithm is surprisingly robust to non-uniform densities, delivering good coverage for almost all nodes.

For the simulations used in this section, the adjustment interval, neighbor expiration, and timing of push and pull messages are all the same as in Section III.

1) Uniform Density: Table I shows the parameter space we explored for investigating the impact of widely differing initial range settings for uniform networks. We generalize the robustness evaluation by varying the ratio of initial average degree to the degree in the ideal R_o case. (The degree is computed as the expected number of nodes within the transmit circle defined by range R.) Using degree alleviates us from

X span(m)	Y span(m)	Z span(m)	number of nodes	λ_p(pkts/sec)	Max. Range(m)	$R_{start}(m)$
7.62	13.4112	8.1534	3162	0.6	1.9	1.0

TABLE II
Non-uniform density network parameters.



(a) Average radius over time (b) Final range histogram (c) Histogram for nodes (out) degree. (d) Coverage histogram.

Fig. 5. *Non-Uniform Density Results.*

performing many experiments testing the 2-dimensional space of λ_s and λ_p. Degree variance is a reasonable generalization, since [5] has shown that the critical factor for coverage is the number of nodes in range.

Our experiments can thus be categorized by the starting degree away from the ideal degree. Our 6 experiments fall into those where the nodes start from too short a range, and thus too small a degree (experiments 1, 3 and 5), and too long a range and a resulting too high degree (experiments 2, 4 and 6). Table I shows that our algorithm must be robust to degrees varying by 2 orders of magnitude. In addition, the sending rate scales from very low, 0.06 pkts/sec, to fairly high, 2.0 pkts/sec. All the experiments correspond to 4374 nodes in a $90 \times 90 \times 90\, m^3$ space and the maximum range is set to $20\, m$. The network bandwidth is set to 10 Kb/s and packet size is 50 bytes, yielding a T of 40ms.

a) Convergence Speed: Figure 2 (a) and (b) show the convergence speed of our algorithm. They show the average range R as a function of the adjustment interval. Figure 2(a) corresponds to the convergence time for a too low initial degree and Figure 2(b) for a too high initial degree. Note that convergence is achieved in most cases in less than 15 adjustment intervals; as mentioned in Section III-B, in the worst scenario (where nodes all start with extremely high range settings) it took a bit longer to converge. The algorithm is also quite stable; in all cases once the proper range is set it only varies by a fraction too small to be seen.

b) Resulting Ranges and Coverage: Figure 2 (c) and (d) show a histogram of the R_o settings for individual nodes. Three features are of note. First, in all cases the result is fairly smooth with the mean close to those predicted by the model (see Table I). Second, the lower the sending rate, the narrower the curve. So, lower sending rate yields better density estimation. As described in Section III-B, this is because the lower sending rate implies more nodes covered, which gives the transmitter more information to estimate density. Finally, note that the curves between the two degree settings are almost identical. This is because identical networks are used, thus a given node should set its final range the same in either the low or high degree experiment.

Figure 3 shows coverage for our 6 experiments, which is computed as the average coverage starting from adjustment interval 30. We again observe standard-looking bell curves. In addition, notice how closely the curves for a given network configuration and sending rate are to each other even though the nodes start with much different ranges.

c) Randomized Initial Ranges: The results of using random initial settings are shown in Figure 4. Figure 4(a) shows that convergence is achieved in 15 adjustment intervals and is stable around the mean. Figure 4(b) shows that the starting ranges are evenly distributed, and the finishing ranges have converged to close to the mean. More importantly, Figure 4(c) shows nearly the same coverage histogram as the uniform starting condition for the same rate (Figure 3(b)). This final coverage demonstrates our algorithm can converge to a point delivering the predicted coverage whether or not the initial range settings are too short, too long, or randomly distributed.

2) Non-Uniform Density: We also ran experiments to look at the algorithm robustness with respect to non-uniform densities. In order to get a realistic set of non-uniform spatial points to run our simulations we performed a spatial inventory of all objects in a graduate student lab. We recorded the smallest bounding box of the 527 objects in the room. To reduce a bounding box to a single point, we use a corner point of the box and adjust its Z to the mean of the box's Z-coordinates. To test the non-uniform case on a scale closer to the uniform network, we replicated the room 2 times along the y-axis and 3 times along the z-axis for a total of 6 replications of the original room. Table II summarizes the non-uniform network parameters.

Simulation results for the above non-uniform case are shown in Figure 5. Figure 5(a) shows the convergence speed is similar to the uniform case, converging in 15 intervals. Figure 5(b) shows the final ranges are quite unlike the uniform case, exhibiting a distinctive non-bell-shaped distribution. However, Figure 5(c) shows a more meaningful demonstration of our

range control algorithm. With the same starting range of 1m, there is a large variance in the out-degree (the number of reachable nodes), spanning from 2 to 128. After convergence, despite the density variance, most of the nodes have the out-degree at around 15. Finally, Figure 5(d) again shows that the mean resulting coverage follows a classic bell-shaped distribution.

B. Effect on Higher-Layer Services

We demonstrate the utility of our range control to higher-layer services by studying its effect on two such services: an unreliable flooding service and a hop-by-hop localization service. We chose to examine unreliable flooding because it is a critical phase in multiple routing protocols [1], [4]. We examined a hop-by-hop localization service for two reasons. First, the approach has been recently explored as a practical localization method [2]. Second, the impact of range control is unclear. On one hand, decreasing the range increases the number of hops, potentially reducing localization accuracy. On the other hand, better coverage and channel utilization should increase accuracy. Thus, observing the impact of our range control algorithm on such a service is a good litmus test to see if range control would be useful to higher-level protocols.

1) Flooding: In our flooding setup, a small fraction of the nodes act as data sources, initiating floods once every 10 seconds. All nodes re-broadcast a flood message if it was triggered within 20 minutes and they see it for the first time.

Figure 6(a) and (b) show the result for a uniform network, where experiments ran on the same network configuration as in Section IV-A.1 with additional 10 roughly uniform distributed data sources and all nodes start with a 20m range. Figure 6(a) plots the number of received messages per node vs. time, averaged over all nodes in the network; Figure 6(b) plots an alternative view: the average number of nodes that a flood message reaches against the send time of that message, averaged across all the data sources. Clearly, range control helps with collision alleviation and thus makes a huge difference in the performance; The scenario with range control(C) performs around 3 times better over the one without(NC) in both packet reception and flooding coverage. Particularly, with range control, on average flooding reaches almost all of the 4383 nodes.

Figure 6(c) and (d) show the result for a non-uniform network, basically the same non-uniform network as in Section IV-A.2 with additional 6 uniformly distributed data sources. We tested with 3 different initial range settings, both with and without range control. Only one range control result is shown here since all the range control scenarios perform almost the same. Aside from the usual collision effect, as shown in the 1.5m range case, we noticed a very unique aspect of non-uniform networks; the 0.5m range case resulted in a disconnected network which strictly limited flooding. From the flooding coverage point of view, a 1m range setting seems proper; its performance closely approaches that of with range control, but again this is only from a single application's point of view and pre-setting the range to the proper value is not an easy task.

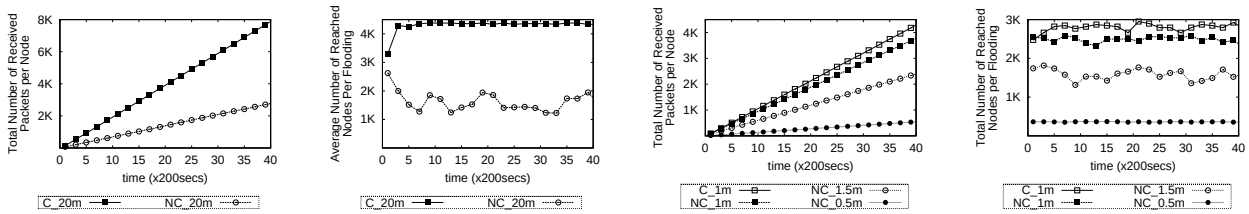
2) Localization: Our localization system is based on the "DV-distance" method proposed in [2]. The approach is to use a small, fixed number of landmark nodes which start with known locations. When these landmarks broadcast their locations, normal (non-landmark) nodes can pick up the messages and compute their distances to the landmarks. Normal nodes can then further distribute the collected information so that nodes can approximately calculate their distance to the landmarks hop by hop. Nodes estimate their positions by trilateration after collecting at least 4 landmark distances. An interesting effect of range on this algorithm is that longer ranges always result in better accuracy because obtaining a better landmark message always improves accuracy. The question for range control then is how it speeds up the convergence if the initial range is set too long or how it can improve accuracy if the range is too short.

In our experiments, landmark nodes keep a constant range and broadcast every 100 seconds; Normal nodes adjust range under range control and propagate information every 3 seconds. Figure 7 shows the localization accuracy over time with and without range control. Each line in the graph represents the percentage of nodes that were able to localize to within a given number of meters to their actual location. Figures 7 (a) and (b) correspond to results in a uniformly dense network. Simulations ran on the same network configuration as in Section IV-A.1 with additional 5 uniformly distributed landmarks and all nodes start with 20m range. The Figures show that range control under uniform densities significantly speeds up the convergence of the algorithm, with little cost in accuracy. We checked the accuracy by slowing down the rate for the no range control case and allowing the algorithm to converge; without slowing the rate down the time to wait for convergence would be much too long. Qualitatively, our simulations show that at long ranges the localization can take hours, as opposed to a few minutes when using range control.

Figures 7 (c) and (d) correspond to results for non-uniform density. Simulations ran on the same non-uniform network as in Section IV-A.2 with additional 6 uniformly distributed landmarks. Landmarks maintain a constant transmission range of 1.9m while normal nodes start with a 1m range. As shown in Section IV-B.1, here 1m range setting in this non-uniform case does not affect the information propagation much; the prominent benefit of range control came from its wider 1-hop information coverage, which gives better approximations for distance calculations and thus better location estimation. With range control, over 60% of the nodes could localize to within 2m while only around 10% of the nodes could reach that accuracy if no range control is employed.

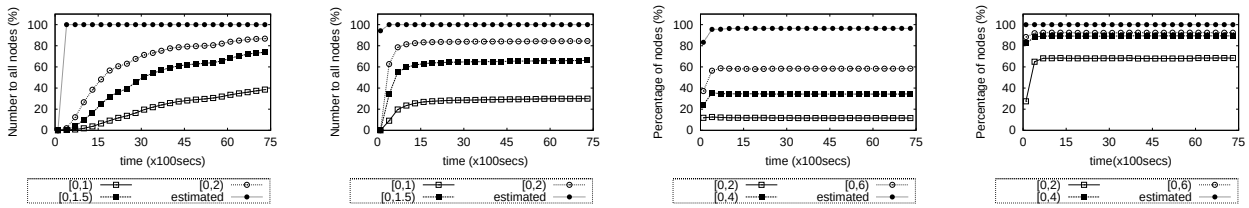
V. Related Work

Our spatial re-use strategy for managing the spectrum in dense networks stands in contrast to the rate or spectrum control taken in other approaches. We do not consider a spectrum control approach because often the networks in these



(a) Uniform (b) Uniform (c) Non-Uniform (d) Non-Uniform

Fig. 6. *Effect of range control on flooding.*



(a) Uniform , no range control (b) Uniform , with range control (c) Non-Uniform , no range control (d) Non-Uniform , with range control

Fig. 7. *Localization accuracy vs. time.*

devices [9], as well as commercial available networks such as 802.11, are limited to a small number of channels. Rather, we view our approach as being complimentary to any approach that uses spectrum control.

Beyond spectrum control, we categorize related work into three general areas: topology control, MAC layer improvements, analytic modeling and intelligent flooding.

a) Topology Control: While all the range adjustment to control topology work below improves spatial reuse, their more prominent goal has been energy conservation while maintaining connectivity. Our work does not consider energy use; instead maximize the 1-hop broadcast coverage at the expense of energy. Our work also does not provide connectivity guarantees.

[10] formulated topology control as an optimization problem and provided centralized algorithm to find the minimum power that should be used by each node to achieve the objective. Their proposed distributed algorithms used node degree and routing connectivity information as heuristic trying to guarantee connectivity. [11] used an orientation approach. They proved that if a node had at least 1 neighbor node within each $2\pi/3$ cone, then the network is as connected as it would have been using maximum power. Their algorithm increases the transmission power until it achieves the desired neighbors per cone. Algorithm [12] works closely with the routing layer. They ran different routing daemons at different power levels and so were able to construct multiple routing tables corresponding to the expended power. By locally comparing the routing tables, a machine could choose the minimum power level needed to forward a packet.

Another method to control the topology is to reduce the number of communication nodes rather than perform power control for all of them [13], [14]. This strategy makes sense for sensor networks because in this context, overlapping sensing regions means not all nodes are needed. Our scheme does not assume the possibility for one node to take up the responsibility of another, i.e., functionality redundancy in the network.

A variety of centralized [15] or distributed scheduling [16] have been used to control the topology for wireless networks. However, we consider centralized approach prohibitive in dense wireless sensor networks. In addition, the above distributed reservation approach relies on feedback, which is difficult to manage for the broadcast messages that we considered here.

A feedback style approach was taken in [17] to optimize throughput, although they used a separate feedback channel. The critical difference between our approach and a feedback style is that we use non-varying parameters to compute the ideal range, so feedback between the dependent (i.e. coverage) and independent (i.e. range) variables is not necessary. However, accurate rate and density observations are critical to our approach.

b) MAC layer improvements: Typical 802.11 MAC layers use RTS and CTS control packets set at maximal power to eliminate the hidden terminal problem. RTS/CTS schemes lowering spatial reuse by blocking possible concurrent transmissions. One scheme [18] proposed to transmit data packets at the minimum required power level thus reducing energy use, however, there was no spatial reuse improvement. Another work [19] proposed a separate busy tone channel, which was used by each node to advertise the additional noise level that could be accommodated and thus could support more concurrent transmissions, i.e., better spatial reuse.

RTS/CTS schemes rely on the interaction between the transmitter and the receiver, which does not work for the broadcast messages we considered here. In comparison, instead of modifying MAC directly, our range control scheme works above MAC layer.

c) Analytic Modeling: The analytic modeling work most related to ours is [6], which found the optimum degree where packet forwarding would have the best expected progress. Our analytic modeling methods are similar in that we reason about the best expected likelihood of packet reception.

d) Intelligent Flooding: A body of work attacks the broadcast storm when performing a flood [20]. Several ideas have been proposed to solve the redundancy, contention and collision problems arising from naive flooding schemes [20], [21], [22]. These mainly use heuristics or local information to eliminate unnecessary rebroadcasts.

Another work took a power control approach [22]. It favors neighbors that can be reached with lower transmission power and nodes further away would be covered by relays from the nearer neighbors; broadcast power is decided based on exchanged local 2-hop neighbor information to maintain coverage.

These approaches specifically target the flooding application. Compared to the above approaches, our 1-hop approach can benefit a broader range of applications, including the ones with different multi-hop objectives.b

VI. Conclusion

In this work we presented a fully distributed range control algorithm for ad-hoc wireless networks to maximize the 1 hop broadcast coverage. Our algorithm is in effect performing distributed topology control on the broadcast packets. Unlike most topology control approaches, however, we took a geometric rather than a graph theoretic approach to reason about the impact of the effects of range on coverage.

We showed via simulations that for starting ranges which are too short, too long, and completely randomized, our algorithm can still converge fairly quickly to the near optimal solution, and remains stable. More importantly, we also showed that our algorithm obtains reasonable coverage and converges quickly for realistic, highly non-uniform networks. This result is quite encouraging because the model analysis was performed on uniform densities; our results show that there is enough local uniformity in realistic situations that simple range control algorithms can be effective.

We demonstrated the effect of our range control algorithm over 2 example higher layer services. For flooding, we observed large increases in both the reception rate and flooding coverage. In a simple localization service, it converged much faster and the accuracy was greatly improved in the non-uniform case.

References

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000.

[2] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, 2001, pp. 2926–2931.

[3] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in *First Symposium on Network Systems Design and Implementation (NSDI)*, Mar. 2004.

[4] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[5] X. Li, T. D. Nguyen, and R. P. Martin, "An Analytic Model Predicting the Optimal Range for Maximizing 1-Hop Broadcast Coverage in Dense Wireless Networks," in *3rd International Conference on AD-HOC Networks & Wireless (ADHOC-NOW)*, July 2004.

[6] T. H. and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Transactions on Communications*, vol. COM-32, no. 3, pp. 246–257, March 1984.

[7] J. Hightower, C. Vakili, G. Borriello, and R. Want, "Design and Calibration of the SpotON Ad-Hoc Location Sensing System, unpublished,," 2001.

[8] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000.

[9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "Systems Architecture Directions for Networked Sensors," in *In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, Nov 2000.

[10] R. Ramanathan and R. Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," in *Proceedings of the 19th Conference of the IEEE Communications Society (INFOCOM)*, 2000, pp. 404–413.

[11] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for wireless multihop ad-hoc networks," in *Proceedings of the 20th Conference of the IEEE Communications Society (INFOCOM)*, 2001, pp. 1388–1397.

[12] V. Kawadia and P. R. Kumar., "Power Control and Clustering in Ad Hoc Networks," in *Proceedings of the 22nd Conference of the IEEE Communications Society (INFOCOM)*, 2003.

[13] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2001, pp. 85–96.

[14] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.

[15] R. Ramanathan and E. L. Lloyd., "Scheduling algorithms for multi-hop radio networks," in *IEEE/ACM Trans. Networking*, April 1993.

[16] C. Zhu and M. S. Corson, "A Five-phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," in *Wireless Networks*, July 2001.

[17] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. K. Dao, "Power management for throughput enhancement in wireless ad-hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2000, pp. 1506–1513.

[18] E.-S. Jung and N. Vaidya., "A Power Control MAC Protocol for Ad Hoc Networks." in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[19] J. Monks, V. Bharghavan, and W. mei W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proceedings of the 20th Conference of the IEEE Communications Society (INFOCOM)*, 2001, pp. 219–228.

[20] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *Proceedings of the 5th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Aug. 1999.

[21] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002, pp. 194–205.

[22] X. Chen, M. Faloutsos, and S. V.Krishnamurthy, "Power Adaptive Broadcasting with Local Information in Ad hoc networks," in *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2003.