Maximizing Broadcast Coverage Using Range Control for Dense Wireless Networks

Richard Martin,

Xiaoyan Li, Thu Nguyen

Department of Computer Science

Rutgers University

May, 2003

Future Building Blocks

- Small complete systems
  - CPU, memory, stable storage, wireless network
- Low cost
  - □ $10
- Low power
  - Devices draw power from the environment
- Small size
  - 1cm^3
- Berkeley Mote is a prototype

Motivation

- Future density
  - At $10, tag most objects
  - At $1 tag everything
  - Lab inventory shows 530 objects in
- Heavy use of broadcast
  - Localization (E.g. Ad-hoc Positioning system)
  - Routing (E.g. Dynamic Source Routing)
  - Management (STEM)
  - Time Synchronization

A Common Pattern

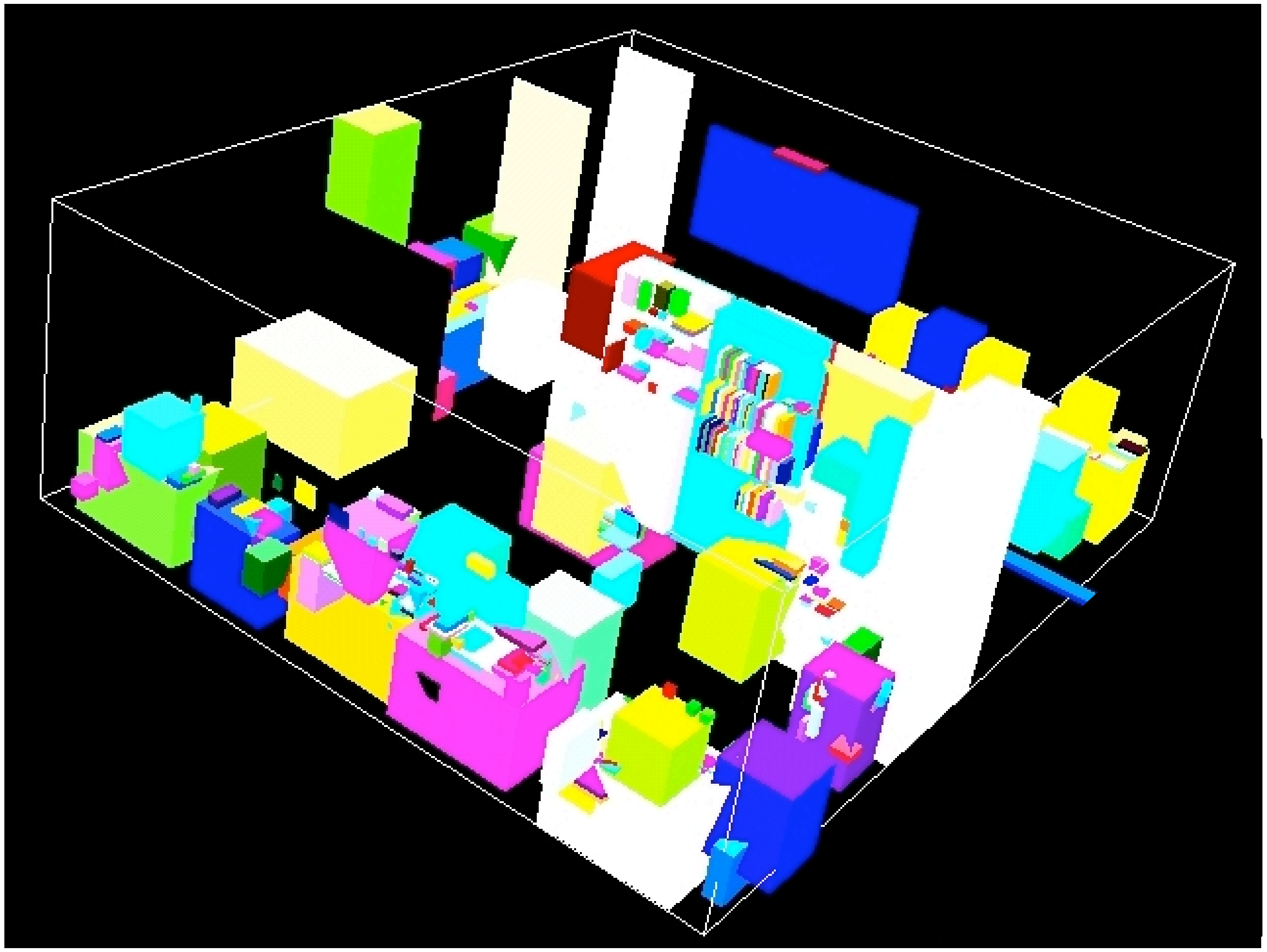
```
Foreach (time-interval) {
    Broadcast(some state);
    Wait(time-interval){
        Collect neighbour responses;
    }
    Do something;
}
```

Spatial Inventory

PANIC Lab

528 objects

137m^3



Problem Statement

- Broadcast, density and CSMA lead to channel collapse
  - Unicast better limits resource using feedback (e.g. RTS/CTS)

- Challenge: maximize number of receivers of a broadcast packet
  - Distributed
  - Low overhead
    - No Extra protocol messages, complex exchanges
  - Fair

Assumptions

- Ad-Hoc Style
- Few channels available
  - E.g. 802.11b -> 11 channels
  - not 1000’s
- CMSA control for broadcasts
- Predictable mapping between range and power

Strategy

- Sharing Strategies:
  - Rate control
  - Channel control
  - Range/power control
- Our approach
  - Passive observation of local density and sending rate to set range to maximize broadcast coverage
  - Set power control to conform to range setting

Implementation Strategy

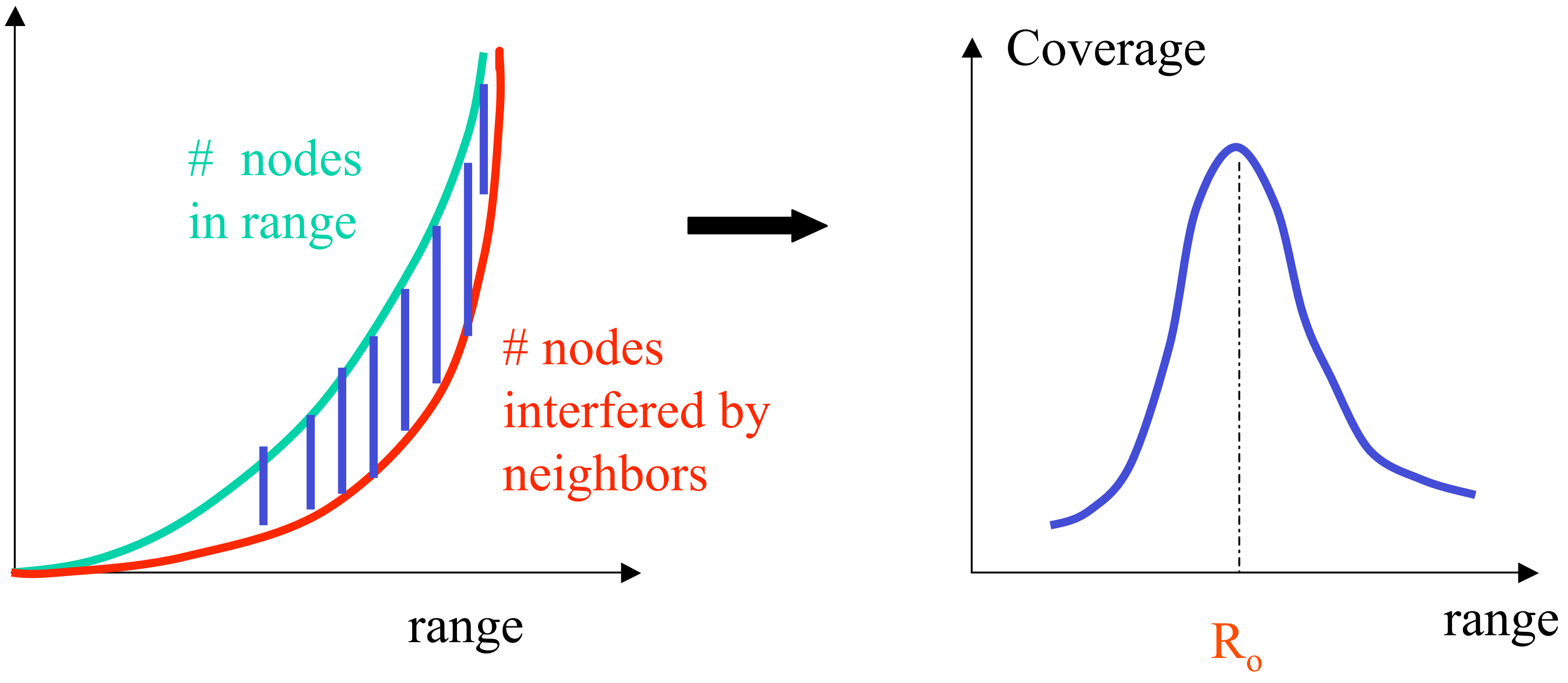
Layer 4	transport
Layer 3	network
Layer 2	LLC
	Ranging & Power
	MAC
Layer 1	Physical

Outline

- Introduction and motivation
- Analytic model of optimal Range
- Application of the model to the distributed algorithm
- Simulation Results
- Future Work and Conclusions

Finding Optimal Coverage

Coverage = # nodes in range – # nodes experiencing interference



Optimal range setting

Analytic Modeling

Want:

- Set range to R_o, which has the highest **expected** coverage.

How:

- Derive a general formula for expected coverage in specific environments and radius setting

  □ **C = f(env, radius)**

- optimal radius is the one which maximize C value

  □ $$\frac{df}{dr}_{(r=Ro)} = 0$$

Analytic Model Basics

- **Node distribution:** multi-dimension poisson distribution: $\square_s$
- **Transmission rate:** poisson packet arrival: $\square_p$
- **Packet Length:** constant size (transmission time **T**)
- **MAC protocol:** CSMA
- **Transmission range:** Nodes use the same radius **R**.
- **Wireless model:**
  - □ Nodes within range R to the transmitter are able to hear the packet.
  - □ More than one transmitter within distance R to the receiver will corrupt all the packets at the receiver.
- **Goal :** Derive the optimal radius setting $\boldsymbol{R_0}$ for specific environment $\{\square_s, \square_p, T\}$

Modeling Inaccuracy

- Mismatch with practical physical transmission model

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{N + \sum_{\substack{k \in \mathcal{T} \\ k \neq i}} \frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta.$$

•No accounting for unicast traffic

- Analytic model inaccuracy:
  - Assume all nodes use the same range
  - Assume transmission times arrive as a poisson process (really CSMA)
  - Geometric approximation

Packet Arrival Simplification

- CSMA makes node transmissions dependent
  - Basically slows down the transmission rate
- Simplification #1
  - assume nodes out of range still follow INDEPENDENT poisson transmission with density
- Effect: Conservative to $\boldsymbol{R_0}$
  - over-estimates the interference coming from neighbors
  - error on side of smaller **R**: prevent channel collapse over more coverage

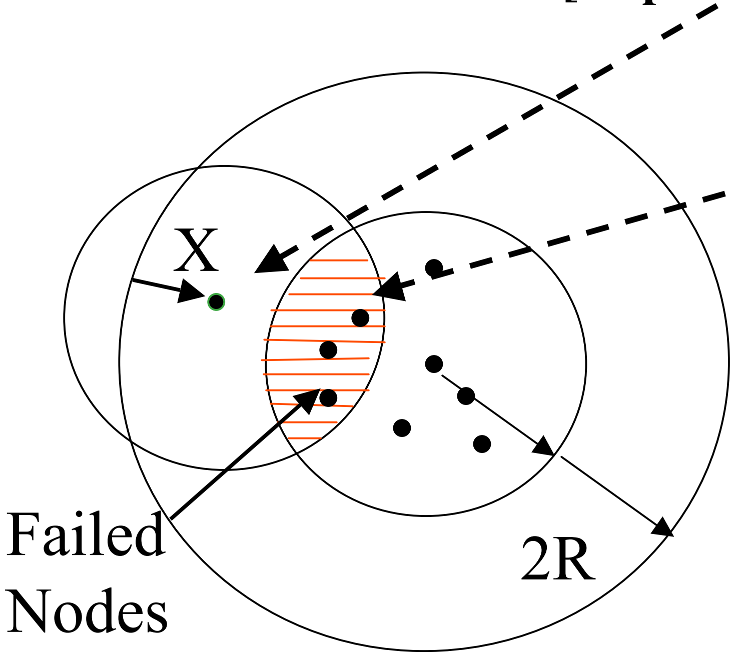
Geometric Approach

- Expected coverage of a packet =

  [Nodes in range]-[losses from hidden terminals]
- Random variable, X, is distance of closest interfering node
  - Compute CDF, I.e. P(x<X)
- Find expected number of failed nodes given at each point in PDF
- Subtract expected number failed from total nodes in range

Geometric Approach

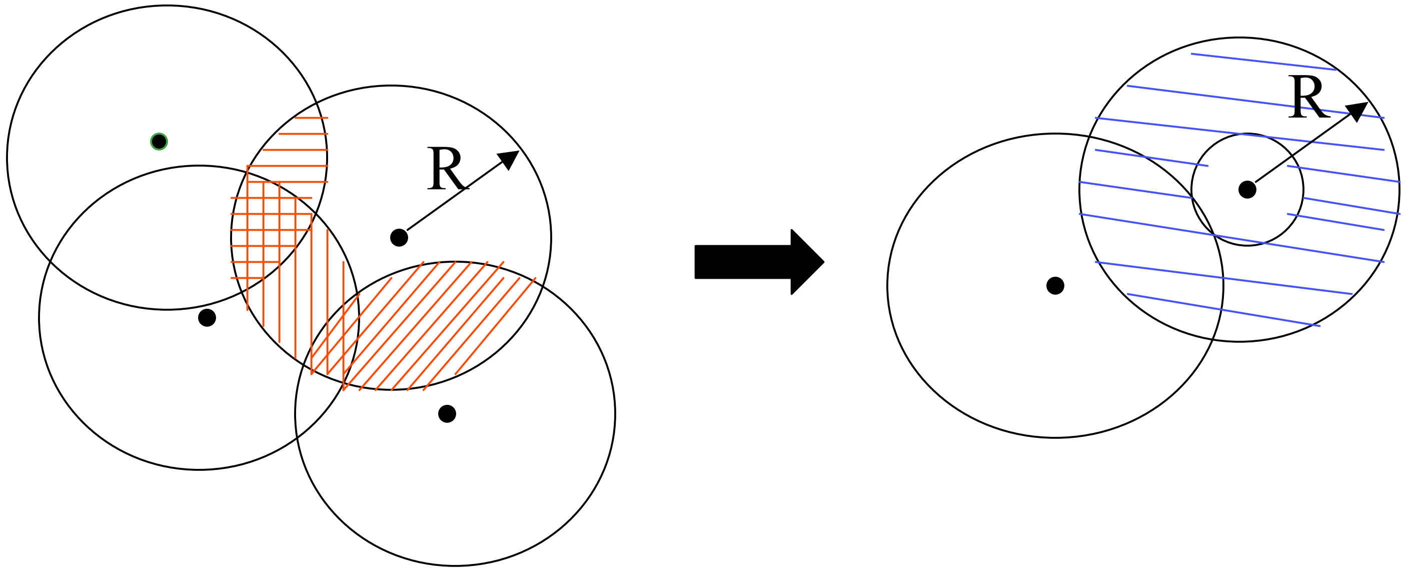
[x= position of interfering node]*[number in affected area]



$E(x) = \int [PDF(x)*(\text{number in affected area})dx$

Geometric Simplification (#2)

Computing expected failing area is difficult



- Torus approximates overlapping intersecting circles(spheres)
  i.e. blue approximates area red.
- This simplification is also conservative to $\boldsymbol{R_0}$

Expected Coverage

CDF (x) = $e^{-\lambda_s * \pi * ((2R-x)^2 - R^2) * (1 - e^{-\lambda_p * 2T})}$

$$E(C) = \lambda_s * (\pi * R^2) - \int_0^R \left[\lambda_s * \pi * (R^2 - (R-x)^2) * P(z \le x \le z + dz)\right] dz$$

Expected nodes in range

Expected number failed

Problem:

- It’s not a closed form formula – can’t solve the integral

   Can’t solve for R_0 directly

Extrapolate to find optimal

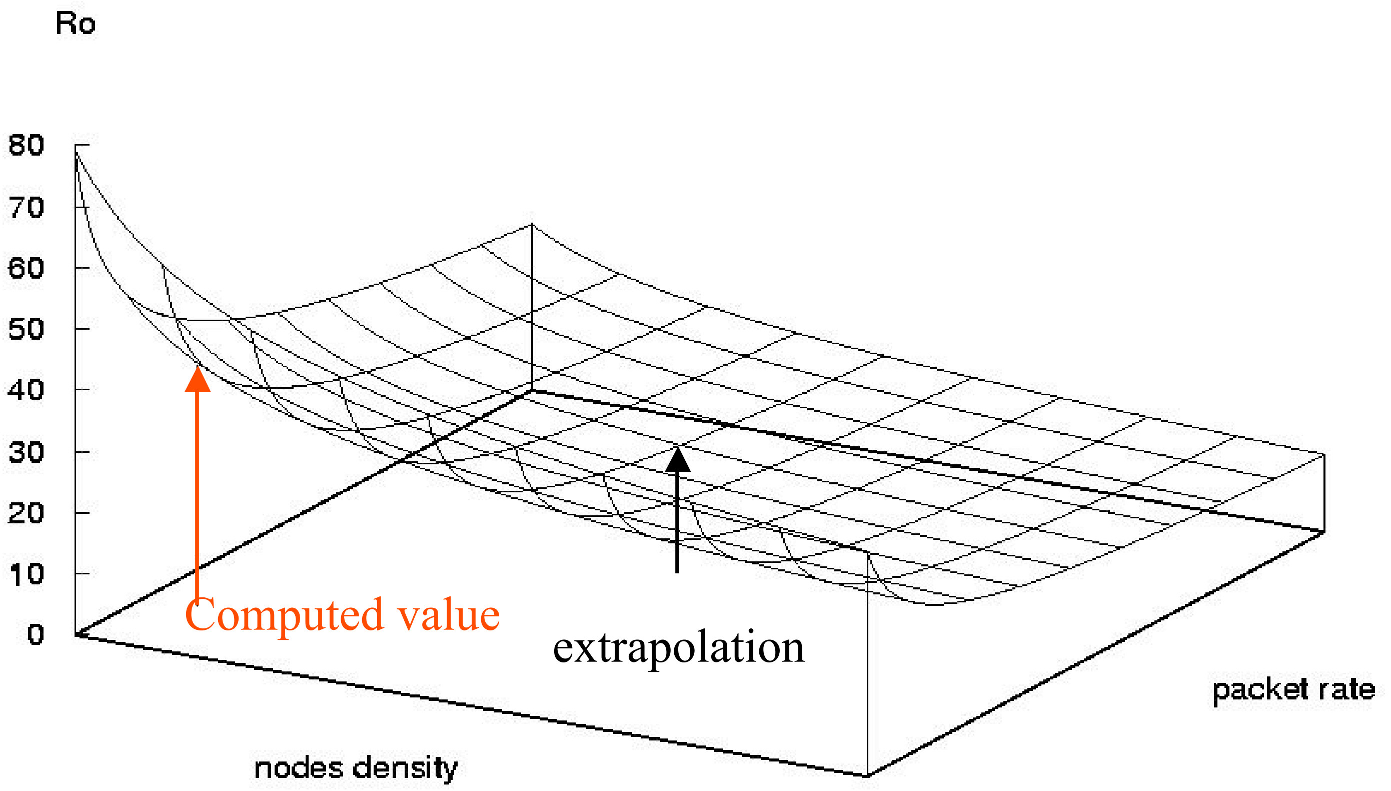
•Solve $\boldsymbol{R_0}$ for the in a specific setting {□$_s$, □$_p$, T} numerically (e.g. maple).

• Assume T is stable – constant packet size.

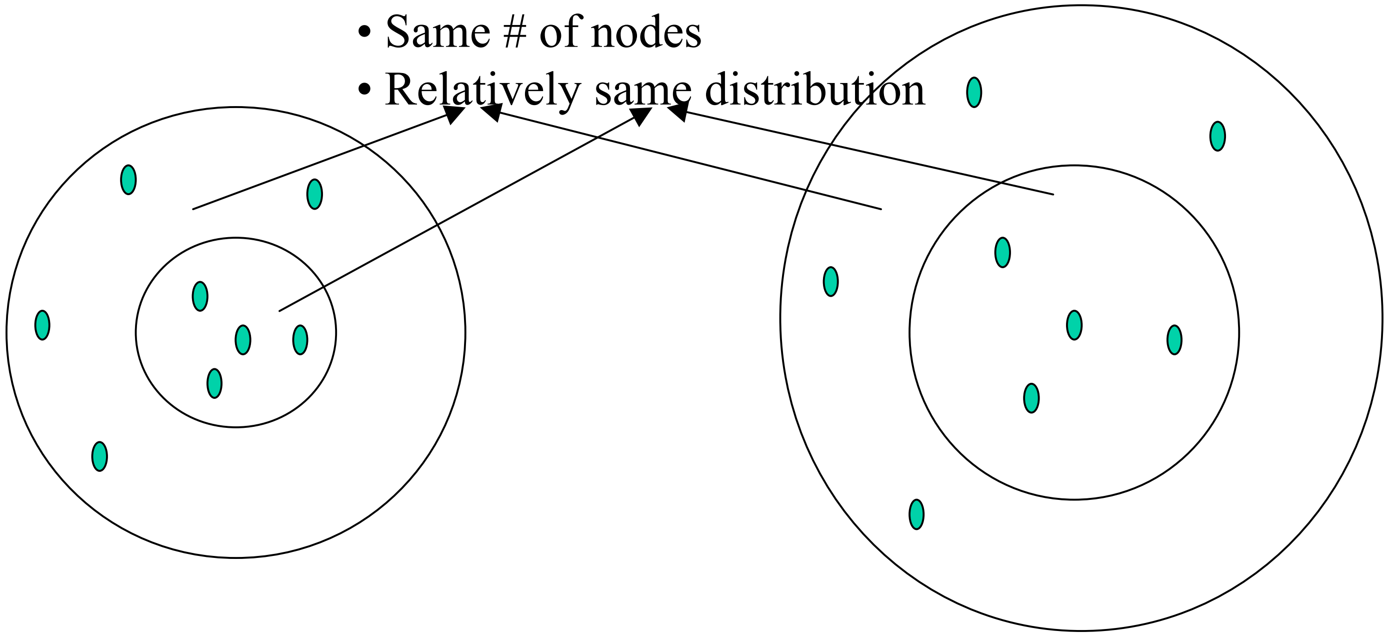
• If we can extrapolate $\boldsymbol{R_0}$ for any arbitrary setting of environments from a known optimal, then we can still apply our idea.

□ $\{\lambda_s, \lambda_p, R_o\} \longrightarrow \left\{\lambda_s', \lambda_p', R_o'\right\}$

Using extrapolations



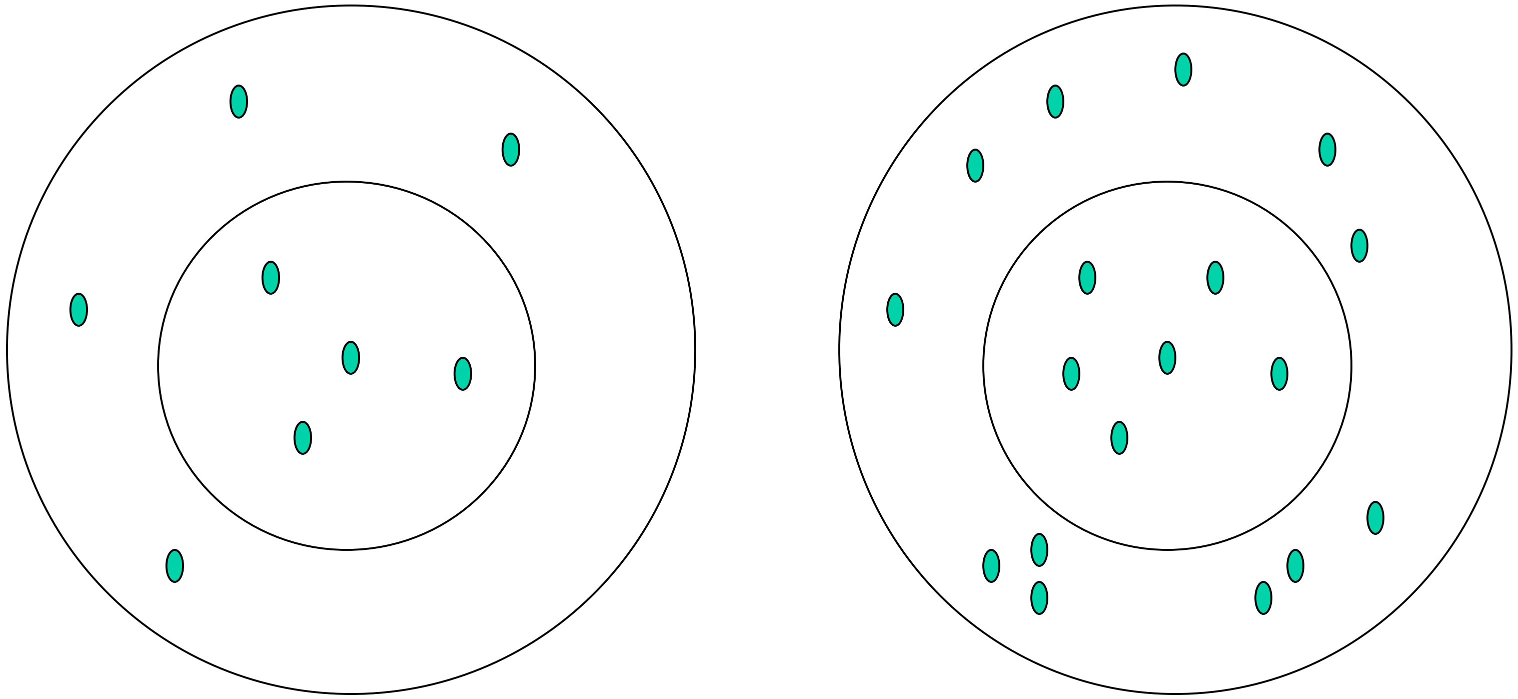
Extrapolation I: Constant Shape



Same rate, different density
Alter R to obtain same # of expected nodes in circle and torus
=> Same expected coverage.

$$\lambda_{p1} = \lambda_{p2}\ ,\ \lambda_{s1} * R_1^2 = \lambda_{s2} * R_2^2 \quad \Rightarrow \quad E(C_1) = E(C_2)$$

Extrapolation II: Constant Packet Volume



Fewer nodes sending frequently is equivalent to more nodes sending infrequently

$$\lambda_{s1} * (1 - e^{-\lambda_{p1} * 2T}) = \lambda_{s2} * (1 - e^{-\lambda_{p2} * 2T}), R_1 = R_2 \quad \Rightarrow \frac{E(C_1)}{E(C_2)} = \frac{\lambda_{s1}}{\lambda_{s2}}$$

Extrapolation accuracy

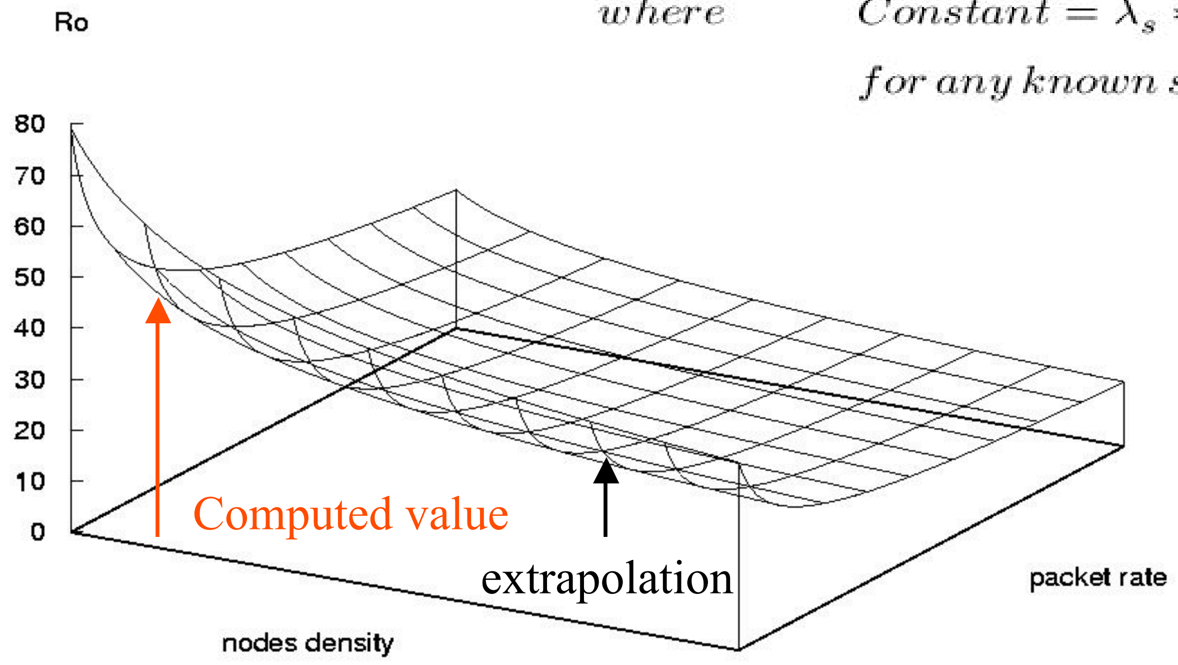
- Extrapolation I (spatial) is exact
- Extrapolation II (network volume) is approximate
  - assume nodes’ transmissions are still independent in spite of CSMA
  - More nodes, more collisions
  - Higher density, less collisions
  - Not clear which effect is stronger

Combining Extrapolations

$$R_o = \sqrt{\frac{Constant}{\lambda_s * (1 - e^{-\lambda_p * 2T})}}$$

$$where \quad Constant = \lambda_s' * (1 - e^{-\lambda_p' * 2T}) * R_o'^2$$

$$for\ any\ known\ set\ \{\lambda_s', \lambda_p', R_o'\}$$



Verification of extrapolations

Node density (nodes/m^3)	Packet rate (pkts/sec)	R_o extrapolation error
0.002 – 0.01	0.4 – 2.0	11% - 20%

Conservative assumptions:

- constant fudge factor of +5% ”safe”

The Distributed Algorithm

- Over an adjustment interval
  (20 broadcasts)
  - Collect neighbor list
    - Neighbors expire if not refreshed for 5 intervals
  - Average send rate
- Compute density at end of interval
  - Use assume spheres
- Set R_o for the next interval
- If only it were that easy ...

Handling Imprecision

- Analytic model assumes perfect information
- Approaches to handling imprecision:
  - Warm up period
  - Overload/underload disambiguation
  - Outlier consideration
    - Minimize impact of outliers
  - Longer-range push and pull messages
    - Insure accurate density estimates
    - Accounts for non-uniform densities

Initialization/warm up

- Initial guess of R
- Wait at least one interval
- Adjust R until there are sufficient neighbors (N)
- If the channel is in overload:
  - Reduce R to cover half the volume
- If not enough expected nodes based on density (underload):
  - Increase R to double volume
  - Expected N = $\pi \frac{C_o}{(1-e^{-\lambda_p 2T})})$
- Once neighbor list is >=N, set $\boldsymbol{R_0}$
  - continue to set each interval based only on last desity and rate

Outliers

- Keep outliers from impacting local density estimate
- Use median
  - Sort neighbours based on distance
  - Keep a running density computation
  - Take median density

Increasing accuracy with extended range messages

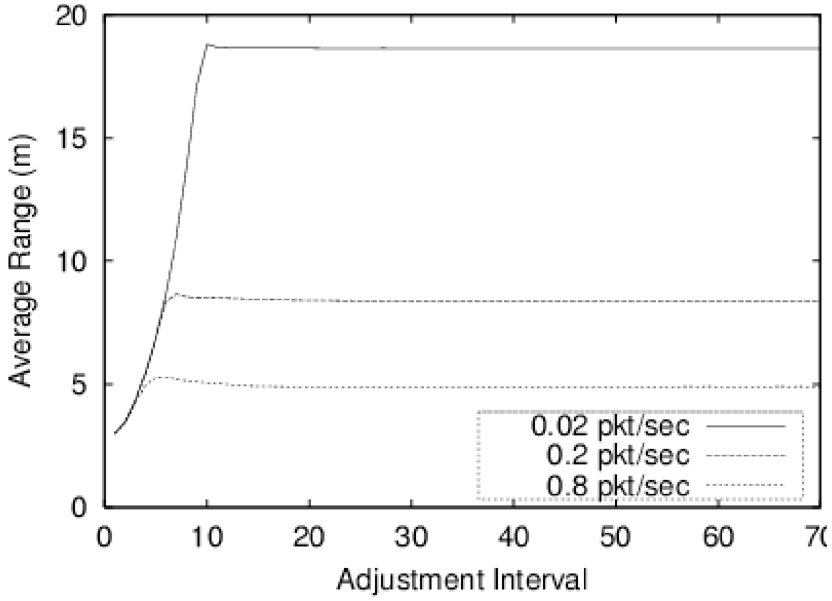
- Pull and Push messages
  - just extend range of a normal broadcast
- Pulls account for hidden terminals
  - Density estimate should include hidden terminals
  - Range set to 2x volume
- Pushes account for asymmetric ranges
  - Nodes should account for all affected nodes
  - Range set to distance of furthest node
  - Accounts for non-unform densities
- 2% of broadcasts are push or pulls
  - Neighbors from push/pull expire after 25 intervals

Simulation Results

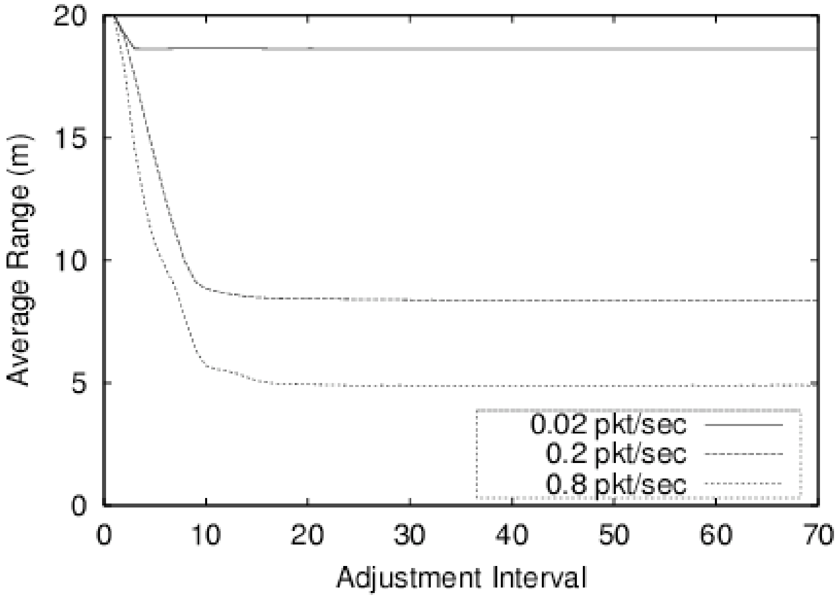
- Simulated 3-D environment
  - Simulations of 5K nodes, 100m^3
- Tested robustness to initial conditions
  - Ranges too high, too low, random
  - Observe convergence speed, final ranges and coverage
- Tested robustness to non-uniform density
  - Used topology based on lab inventory
- Observed impact on a higher-level protocol

  A hop-by-hop localization protocol

Convergence speed

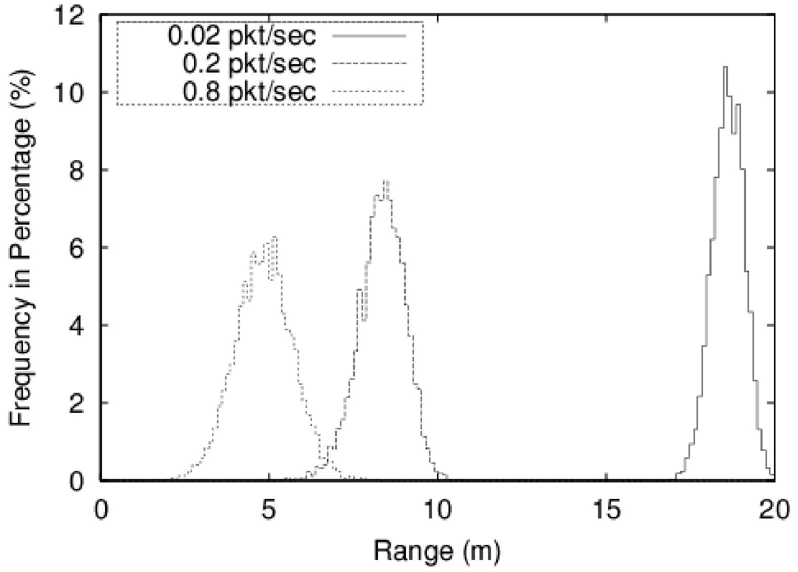


Initial R=3

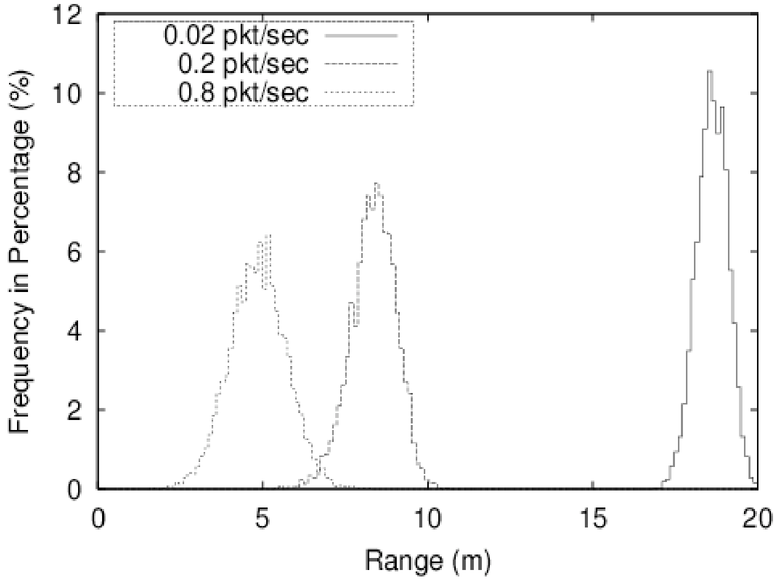


Initial R=20

Robustness to Initial Ranges

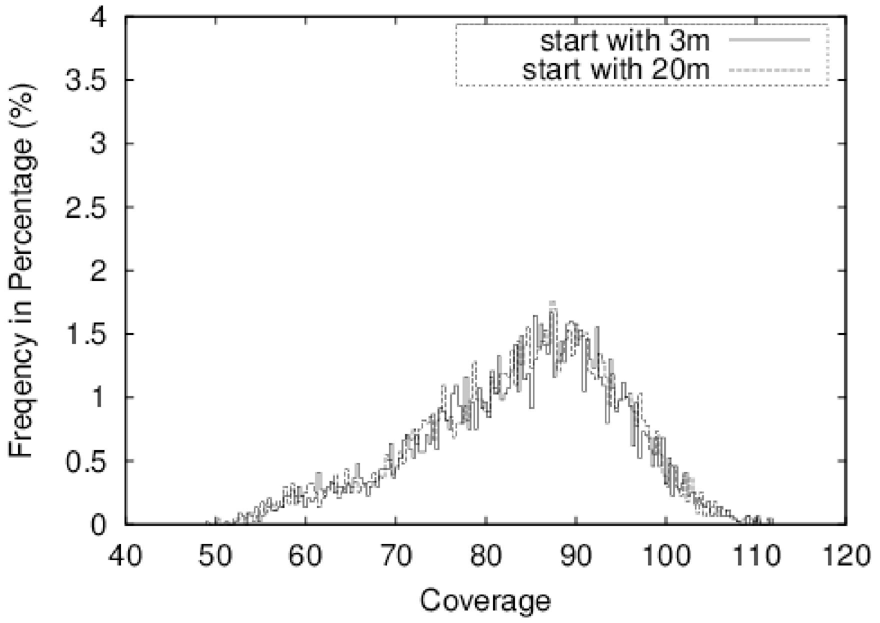


Initial R=3

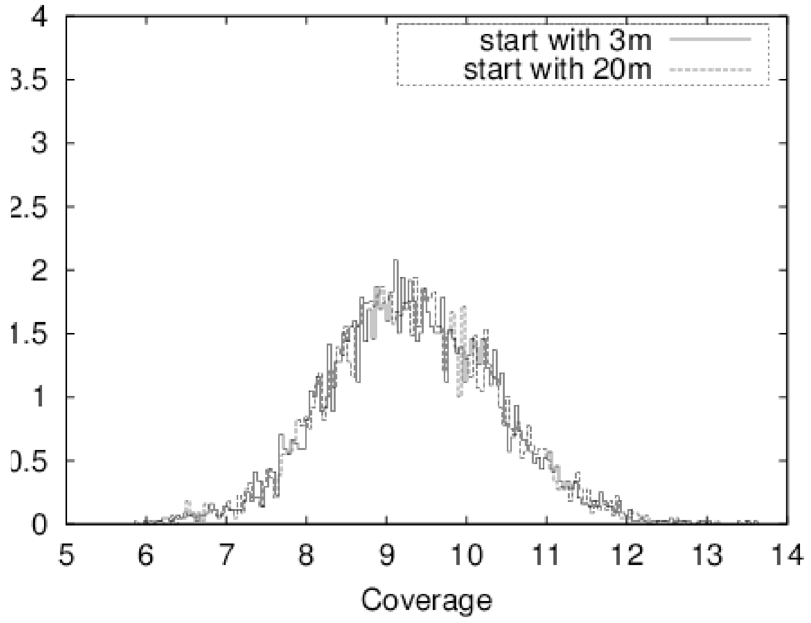


Initial R=20

Final Coverages

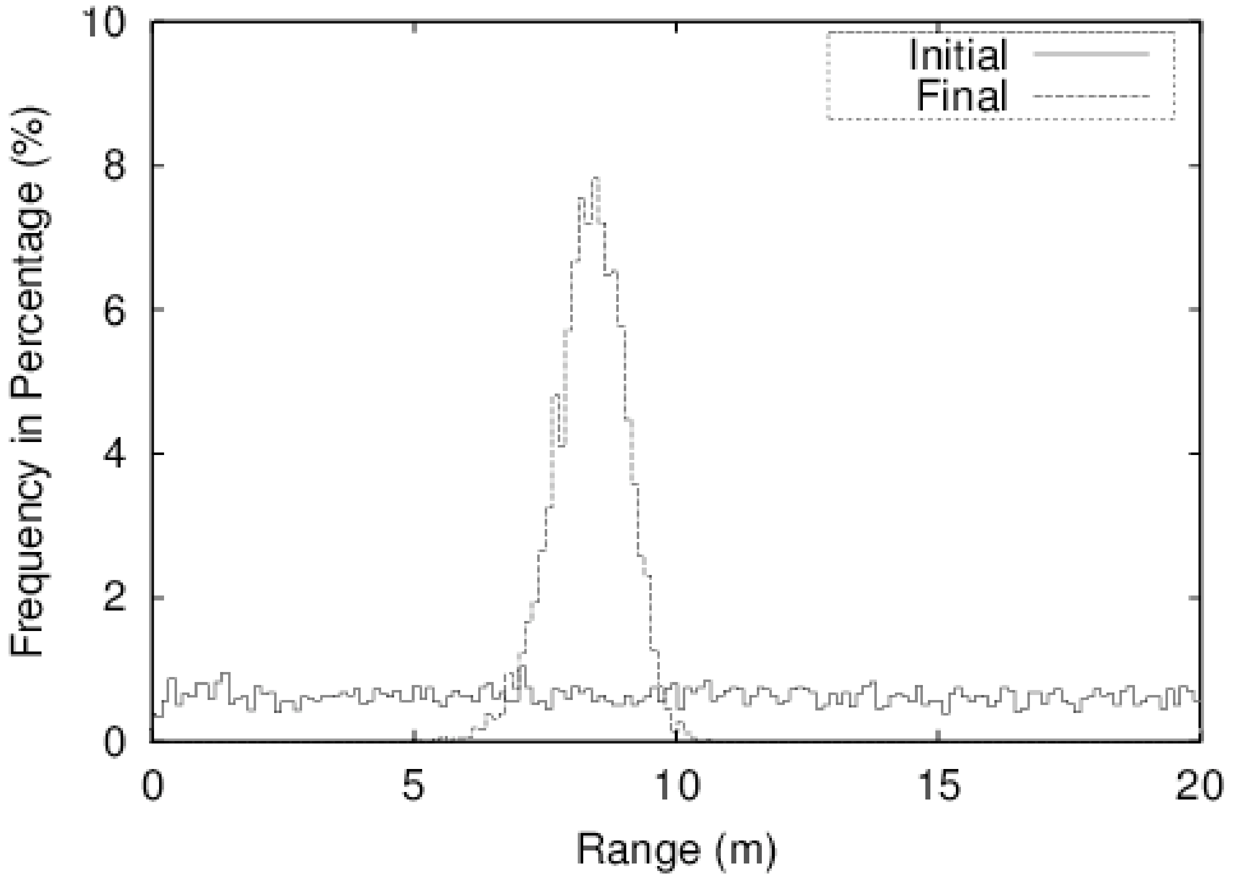


Arrival Rate =0.02 pkts/s



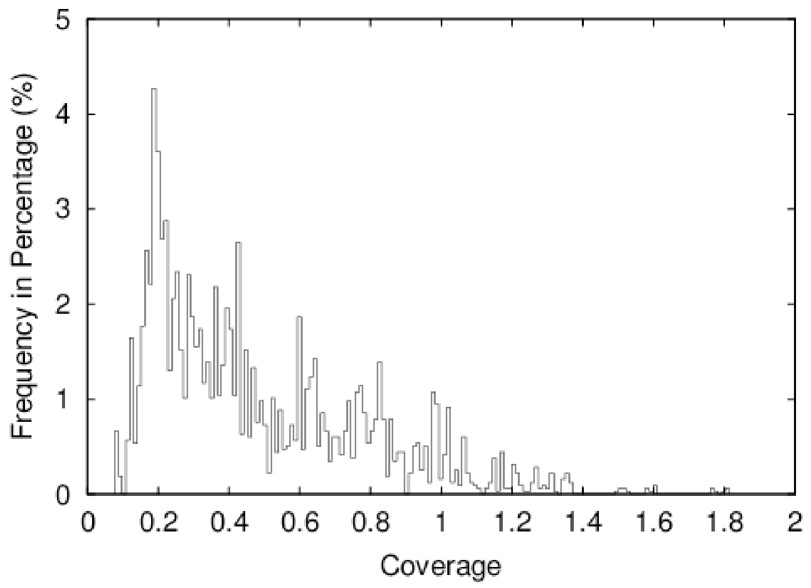
Arrival Rate =0.2 pkts/s

Robustness to Random Initial Ranges

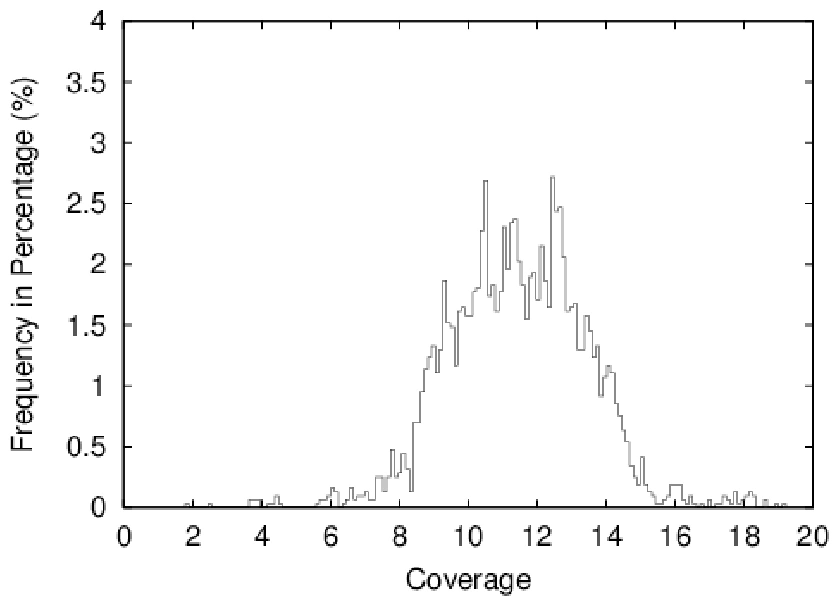


Non-uniform networks

Initial coverage



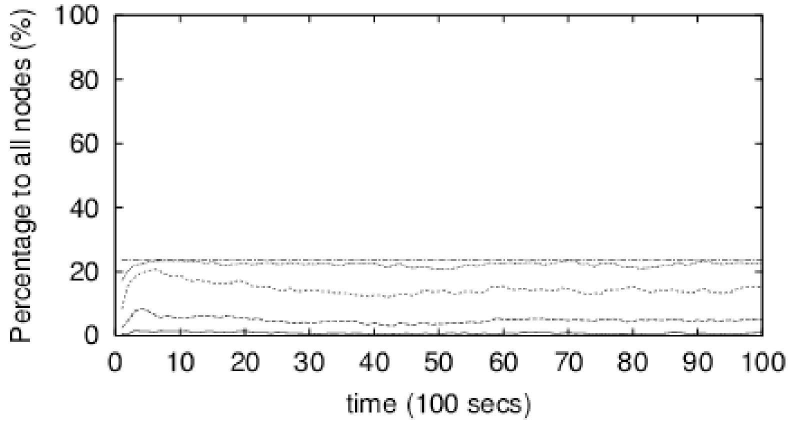
Final coverage



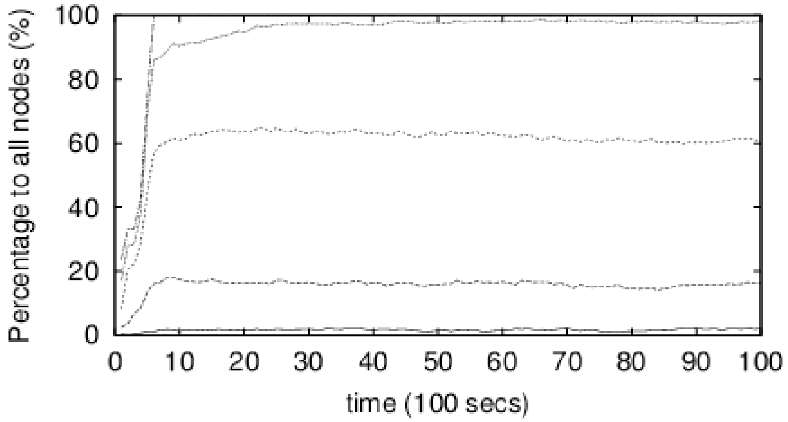
3100 nodes (lab replicated 6x),

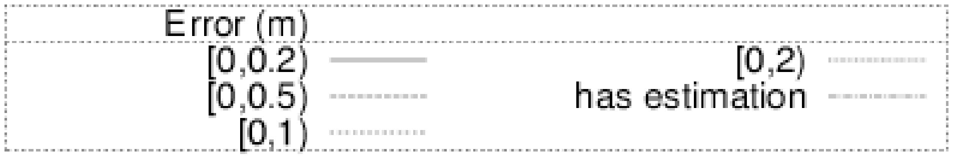
Impact on a localization protocol

No Range Control



Using Range Control





Future work and Conclusions

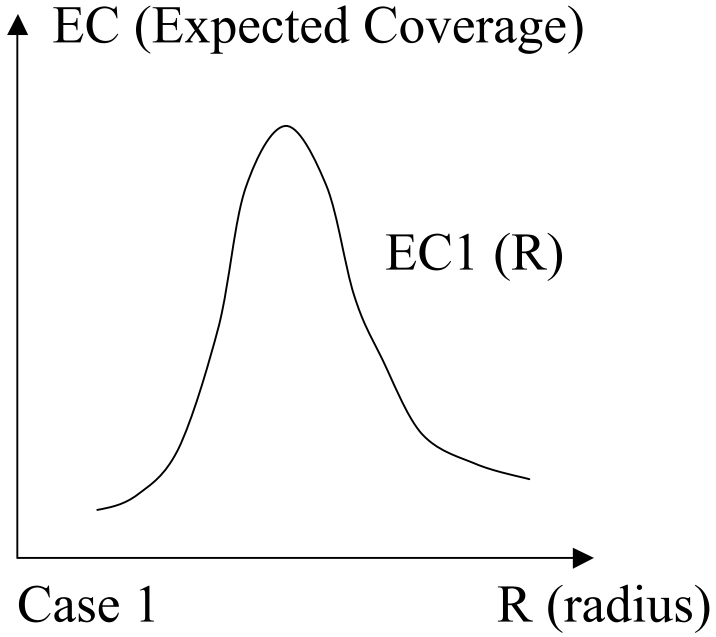
- Range control promising approach
- Continue validations:
  - Floor and building-wide simulations
  - Dynamic Network (join and leave)
  - Real implementations
    - 802.11 and motes
- Need more higher-level protocols
- Need realistic traffic patterns
  - Chicken and egg problem

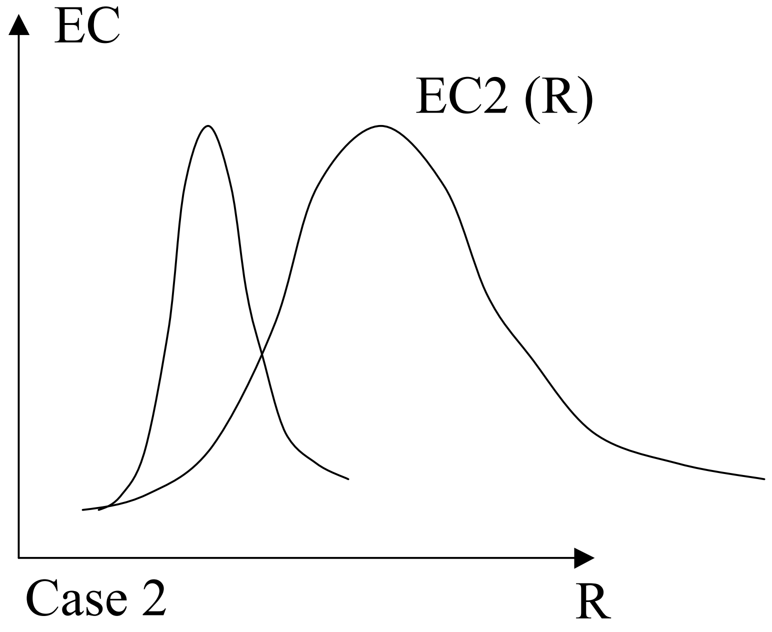
Backup slides

- These slides are for questions and answers

Extrapolation based on rule I

$$\lambda_{p1} = \lambda_{p2}\ ,\ \lambda_{s1} * R_1^2 = \lambda_{s2} * R_2^2 \quad \Rightarrow \quad E(C_1) = E(C_2)$$



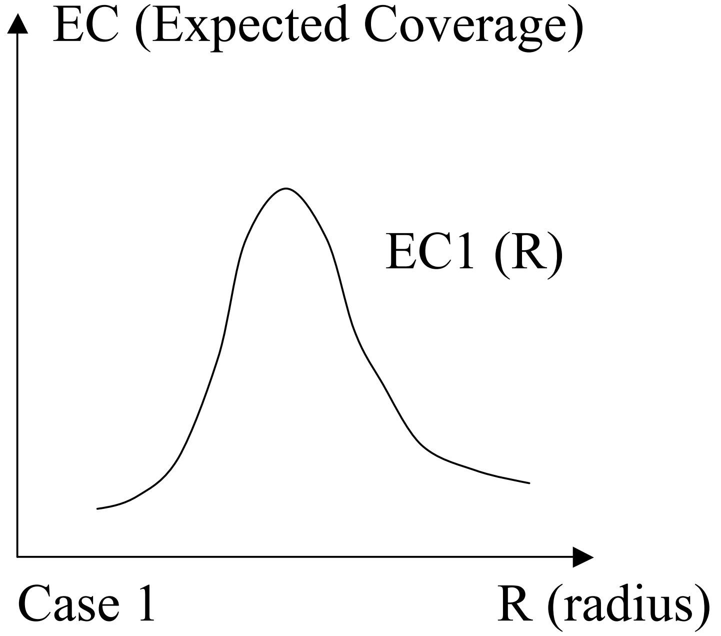


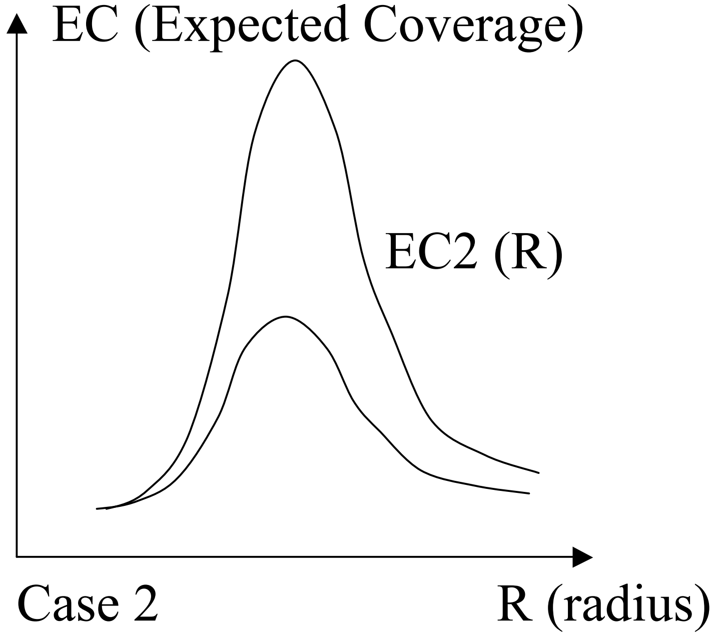
$$\lambda_{p1} = \lambda_{p2} \Rightarrow EC2(R) = EC1(\sqrt{\frac{\lambda_{s2}}{\lambda_{s1}}} * R)$$

$$\lambda_p = \lambda_p' \quad \Rightarrow \quad R_o' = \sqrt{\frac{\lambda_s * R_o^2}{\lambda_s'}}$$

Extrapolation based on rule II

$$\lambda_{s1} * (1 - e^{-\lambda_{p1}*2T}) = \lambda_{s2} * (1 - e^{-\lambda_{p2}*2T}), R_1 = R_2 \quad \Rightarrow \frac{E(C_1)}{E(C_2)} = \frac{\lambda_{s1}}{\lambda_{s2}}$$

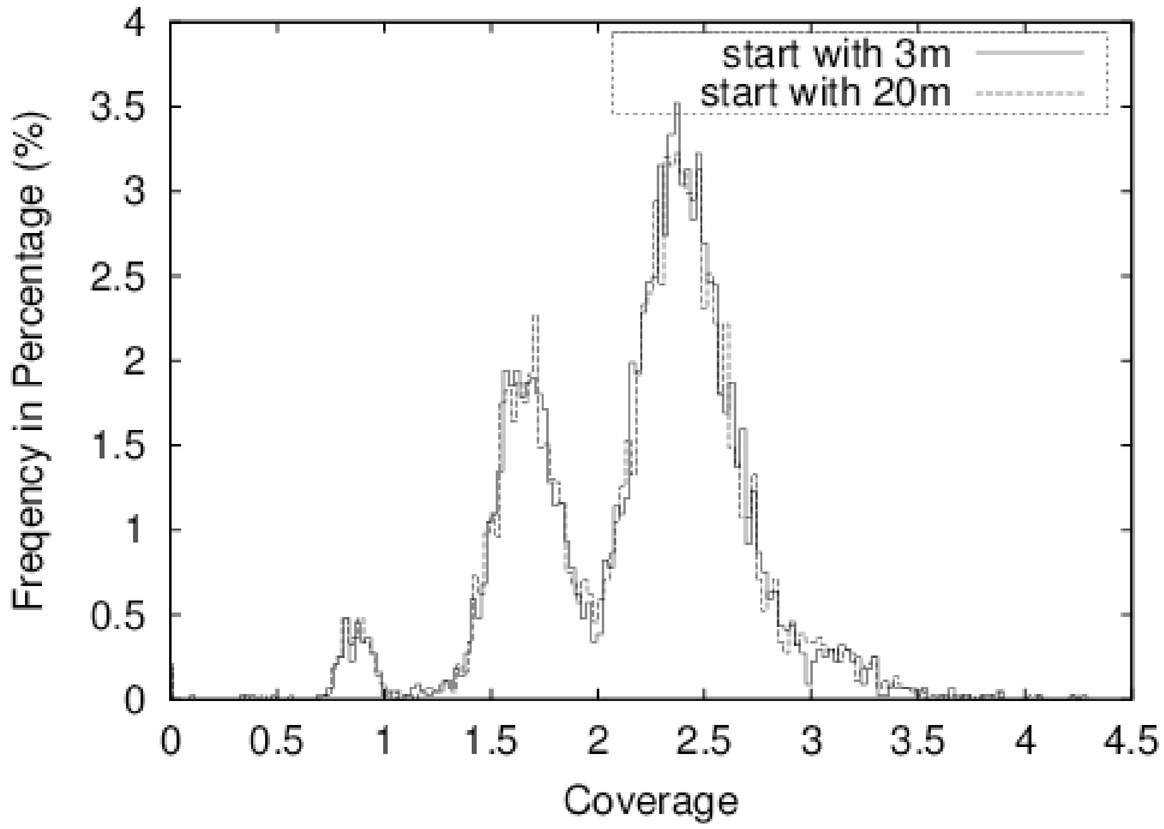




$$\lambda_{s1} * (1 - e^{-\lambda_{p1}*2T}) = \lambda_{s2} * (1 - e^{-\lambda_{p2}*2T}) \Rightarrow EC2(R) = \frac{\lambda_{s2}}{\lambda_{s1}} * EC1(R)$$

$$\lambda_s * (1 - e^{-\lambda_p*2T}) = \lambda_s' * (1 - e^{-\lambda_p'*2T}) \quad \Rightarrow \quad R_o' = R_o$$

Uniform Coverage



Arrivale Rate = 0.8 pskts/sec