5-2020

Re-evaluating coverage metrics for wireless sensor network border security applications

Kevin Haney

5-2020

Re-evaluating coverage metrics for wireless sensor network border security applications

K. Karlton Haney

Re-evaluating coverage metrics for wireless sensor network border security applications

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Industrial Engineering

by

K. Karlton Haney

February 2020
University of Arkansas

Thesis Advisor: Dr. Kelly Sullivan
Thesis Reader: Dr. Chase Rainwater

Acknowledgements

First and foremost, I would like to thank Dr. Kelly Sullivan for his dedication and committtment to this research and his guidance throughout this journey. Next, I would like to thank the Honors College at the University of Arkansas for funding this research through the Honors College Research Grant. Lastly, I would like to thank my family for their support during this process. Thank you mom and dad for encouraging me to take advantage of educational opportunities such as this and for providing me with the resources to do so, such as the computer on which the research was conducted and documented. Thank you Rikki for being patient with me while I spent many long days and nights on this work.

Abstract

Wireless sensor networks are an emerging technology used to monitor an environment over time. One specific application of the wireless sensor network is the border security application. Researchers have adapted coverage metrics from general wireless sensor network literature to fit the border security application. While some of the adapted metrics count the number of sensors detecting a potential target, others measure the distance between a potential target and its nearest sensor. No existing metric accounts for both of these factors. To take advantage of this gap and to attempt to increase the accuracy to which coverage is measured in this application, we created the total intersected length metric.

To test the adapted metrics in comparison to our new metric, we created a MATLAB program to simulate a wireless sensor network over time as its sensors fail and its coverage degrades. Through experimentation with this model on a notional data set, we can conclude that the total intersected length metric used for measuring network coverage for border security WSN applications has both advantages and disadvantages when compared to traditional coverage metrics. The new metric was found to be advantageous for effectively capturing the initial coverage of a network and for determining when a network is no longer sufficiently covered while it was found to be disadvantageous for conservatively capturing network degradation and for producing quick computation times.

Table of Contents

1. Introduction

Wireless sensor networks (WSN) are an emerging technology used to monitor an environment over time. WSN applications include environmental, industrial, and health monitoring; and animal, human, and vehicle tracking [1], [2]. One forthcoming application is in the "internet of things," a network of appliances and devices made smart with the addition of sensor nodes and an internet connection [3], [4].

A WSN consists of a web of many sensor nodes that collect data and transmit it to one or more sink nodes that receive, compile, and process the data before transmitting it to a base station [1], [2], [5]. Advanced sensor nodes can detect stimulants such as motion, noise, speed, and the size of a passing object [3]. Each node is equipped with sensors, short-range wireless communication devices, a processor, and an energy source [2], [3], [4], [6]. WSNs can be quickly deployed over large geographical regions, often by plane or missile. This unique characteristic allows for applications in inaccessible terrains and makes WSNs perfect for military target tracking and security surveillance applications [1], [3]. WSNs also possess self-organizing and self-reconfiguring capabilities to account for random deployment and changes in the network [1], [3], [4], [6].

Each node has a limited range of detection. The maximum distance a node can detect is called the sensor's sensing radius, while the area within the sensing radius is called the coverage disk, see Figure 1.

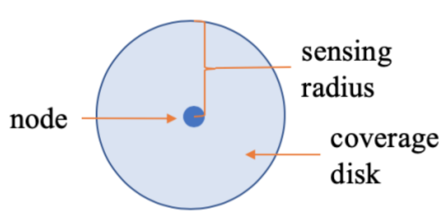


Figure 1: The diagram of a node, including its sensing radius and coverage disk

One of the major challenges associated with WSNs is unreliability. Unreliability is a problem because as time passes, sensor nodes are increasingly prone to failure. The most common reason for failure is the depletion of battery supply due to expending energy to communicate and collect information. Once the battery is fully consumed, nodes are rendered useless [1], [3], [4], [5]. Because WSNs are often located among extreme terrain or battlefields, simply replacing low batteries is not feasible and redeployment is costly [5], [6], [7]. Many WSN applications require effective coverage for months or years [7]; however, many sensor nodes will likely fail during this time [3], [4]. This disconnect is called the reliability or fault tolerance issue which works to maintain overall network effectiveness as nodes fail over time [3].

This research is motivated by the border security application in which the WSN is used to create a barrier between two regions. To relate our work to the "barrier coverage" literature, we now summarise our basic modeling constructs. We assume the WSN is deployed into a rectangular border region. The horizontal coordinates of this region reprensent different points along the border along which a potential threat might seek to cross the border. The vertical coordinate represents the depth of the border region, i.e., the distance a potential threat would have to move (undetected) in order to cross the border successfully. Rather than try to model all paths through the region, refine our attention to a discrete set of *evasion paths* in which the threat moves (in the direction of the vertical coordinate) directly across the border region (see Figure 2). We examine metrics for quantifying the extent to which a WSN covers a given evasion path. Because an intelligent threat (e.g. a smuggler or attacker) might seek an evasion path that exploits an area , we summarize the WSN's coverage with respect to the least-covered evasion path.

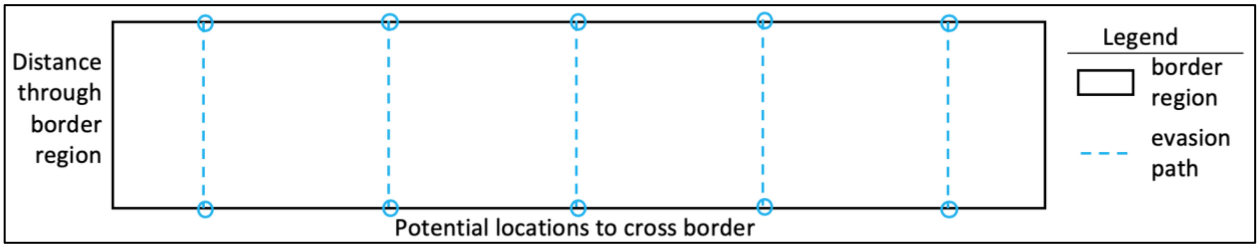


Figure 2: Five evasion paths through border region

In what follows, we summarize relevant barrier coverage metrics within the context of the modeling construct summarized above. Meguerdichian and Koushanfar employ a minimum-distance metric that quantifies (under our modeling constructs) an evasion path's coverage as the distance to its nearest sensor node [8]. Saipulla, Westphal, Liu, and Wang adjust the *k*-coverage technique for barrier coverage, quantifying coverage of an evasion path according to the number of sensors whose coverage disks intersect the evasion path [9]. The minimum-distance metric is believed to consider the distance of the detecting sensor but fail to consider the number of separate detections. The *k*-coverage metric is believed to consider the number of separate detections but fail to account for the distance between a sensor and a security threat. To take advantage of this gap and to attempt to increase the accuracy to which coverage is measured in this application, we created the total intersected length metric. We believe the total intersected length metric accounts for both the distance of the detection and the number of separate detections by combining both considerations into one measurement. To the best of our knowledge, no existing research quantifies coverage of an evasion path as the sum of the regions along the evasion path that are detected by a live sensor node.

2. Methodology

This section formalizes the three coverage metrics (total intersected length, *k*-coverage, and minimum distance) and our simulation model. Section 2.1 summarizes notation and assumptions for our model and metrics. Section 2.2 defines coverage metrics, and Section 2.3 describes the simulation model. Section 2.4 illustrates the methodology for an example WSN modeled according to the specifications detailed in previous sections.

2.1 Notation and Assumptions

The model includes an initial network, a border region, sensors failing over time, a network evolving, and connectivity, as detailed in Figure 3. Line 1 defines the set of nodes *N*(0) at time 0, where x_i is the x-coordinate of node $i \in N(0)$ and y_i is the y-coordinate of node $i \in N(0)$. Line 2 defines the set of edges (between pairs of communicating sensors) at time 0, where r_c equals the maximum communication distance between nodes and d[•,•] is the Euelidean distance function. In line 3, the border region is defines as the rectangle with corners at (0,0), (a,0), (0,b), (a,b). Thus, *a* is the horizontal distance of the border region and *b* is the vertical distance of the border region. Line 5 defines the x-coordinate of each evasion path. Only the x-coordinate is necessary to define an evasion path because paths run vertically through the entire evasion path.

Each node's fail time is assumed to be random. We will estimate the average value of each metric by applying Monte Carlo simulation. Thus, in order to define the coverage metrics, we assume a deterministic failure time t_i is given for each node $i \in N(0)\backslash\{0\}$. Let $t_{(i)}$, $i \in N(0)\backslash\{0\}$ denote the sorted sensor failure time, i.e., such that $t_{(0)} < t_{(1)} < t_{(2)} < \cdots < t_{(n)}$. Line 6 defines the set of nodes *N*(*t*) that survive past time *t* > 0, called live nodes. Line 7 defines the time at which the *i*th sensor fails. As nodes fail, corresponding edges fail as well, leaving a smaller set of edges *E*(*t*) still live at time *t*, represented by line 8. Lines 9 defines the set of nodes connected to the sink node. This set can be determined for *t* > 0 by applying a breadth first search algorithm to network (*N*(*t*),*E*(*t*)).

Initial Network
1: $N(0) = \{0, 1, 2, \ldots, n\}$:The set of sensors initially in the network, where node 0 represents the sink node
2: $E(0) = \{\{i, j\}: d[(x_i, y_i), (x_j, y_j)] \leq r, i \in N(0), j \in N(0)\}$

Security Region
3: $B = [0, a] \times [0, b]$:The security region
4: V : the number of evasion paths
5: x_e, $e = 1, 2, \ldots, V$:The x-coordinate of evasion path e

Sensors Fail Over Time
6: t_i, $i \in N(0)\backslash\{0\}$:The failure time of sensor node i
7: $t_{(i)}$, $i \in N(0)\backslash\{0\}$:The time of the ith sensor failure

Network Evolving Over Time
8: $N(t)$, $t > 0$: The set of nodes surviving to time t, i.e., $\{i \in N(0) : t_i > t\}$
9: $E(t)$, $t > 0$: The set of edges for which both adjacent nodes are in $N(t)$

Connectivity
10: $C_i(t) = \begin{cases} 1 \text{ if node } i \text{ is in } N(t) \text{ and connected to node 0 in the network } (N(t), E(t)) \\ 0 \text{ otherwise} \end{cases}$

Figure 3: Representation of a *WSN over time*

As summarized in Section 1, we modeled WSNs with a discrete set of evasion paths running vertically through them. We assume the evasion paths' are equally spaced along the interval [0,a].

Three coverage metrics are defined with respect to the network's state at a given time. We represent coverage at time s by $K_m(t_1, t_2, \ldots, t_n; s)$ where $m \equiv \{1, 2, 3\}$ indicates which metric is used to measure the network's coverage. Section 2.2 provides more details on each metric.

2.2 Coverage Metrics

Each of the three coverage metrics we used – total intersected length, k-coverage, and minimum distance – consider a border region covered under different conditions. When using the total intersected length metric, the border region is considered covered as long as all evasion paths are detected by live sensors. The distance detected along an evasion path is considered the total evasion path length enclosed by live sensors. Similarly, when the k-coverage metric is used, the border region is considered covered as long as all evasion paths are detected live sensors. An evasion path is considered to be detected by a sensor if any portion of the path is enclosed within the sensor's detection radius. Lastly, when the minimum distance metric is used, the border region is considered covered as long as each evasion path detected by live sensors.

1: I^g_{ei} = maximum (g=1) or minimum (g=2) value of y such that $0 \le y \le b$ and $d[(x_e, y), (x_i, y_i)] \le r$ for $e = 1,\ldots,V$ and $i \in N(0)\backslash\{0\}$
2: $I^1_{ei} = I^1_{ei} * C_i(s)$ for $e = 1,\ldots,V$ and $i \in N(0)\backslash\{0\}$
3: $I^2_{ei} = I^2_{ei} * C_i(s)$ for $e = 1,\ldots,V$ and $i \in N(0)\backslash\{0\}$
4: **For** $e = 1: V$
    For $i = 1: n$
        For $k = 1: n$ such that $i \neq k$
            If $I^1_{ek} < I^1_{ei}$ && $I^2_{ek} > I^2_{ei}$
                $I^1_{ek} = 0;$
                $I^2_{ek} = 0;$
            If $I^1_{ek} < I^1_{ei}$ && $I^1_{ek} > I^2_{ei}$
                $I^1_{ek} = I^2_{ei};$
            If $I^2_{ek} < I^1_{ei}$ && $I^2_{ek} > I^2_{ei}$
                $I^2_{ek} = I^1_{ei};$
            If $I^1_{ei} > b$
                $I^1_{ei} = b;$
            If $I^2_{ei} < 0$
                $I^2_{ei} = 0;$
    For $i = 1: n - 1$
        For $k = i + 1: n$
            If $I^1_{ek} == I^1_{ei}$ && $I^2_{ek} == I^2_{ei}$
                $I^1_{ek} = 0;$
                $I^2_{ek} = 0;$
5: I^0_{ei} = matrix of intersected length for $i = 0,\ldots,n$ and $e = 1,\ldots,V$
6: $I^0_{ei} = I^1_{ei} - I^2_{ei}$
7: $K_1(t_1, t_2, \ldots, t_n; s) = \text{Min}\ \{\sum_{i=1}^{n} I^0_{ei} : e = 1,\ldots,V\}$

Figure 5: Representation of the total intersected length model

The total intersected length metric $K_1(t_1, t_2, \ldots, t_n; s)$ is calculated according to the algorithm in Figure 5. The algorithm uses a three-dimensional matrix I^g_{ei} $(g = 1,2; e = 1, \ldots, V; i \in N(0))$. The algorithm first determines (in Line 1) the intersection points I^1_{ei} and I^2_{ei} for each $e = 1, \ldots, V$ and $i \in N(0)\backslash\{0\}$ by finding the maximum and minimum y-value (with $0 \le y \le b$ so that the point remains in the feasible region) such that the point (x_e, y) falls within the coverage disk of node *i*. However, because two sensors' coverage disks may intersect an evasion path at the same point, computing the total intersected length is not as simple as summing up the length of each sensor node's intersected line segment. The strategy of the algorithm is to adjust the intersected points such that no evasion path segment is counted twice in the measurement.

Lines 2 and 3 multiply both I_{ei}^1 and I_{ei}^2 by the connectivity matrix $C_i(s)$, effectively removing failed and/or disconnected sensors from consideration by setting their intersection points to zero. The line segment $[(x_e, I_{ei}^2), (x_e, I_{ei}^1)]$ now describes the set of points on evasion path e = 1,…,*V* that are covered by a connected sensor $i \in N(t)$. Line 4 outlines logic to modify, for a given e = 1,…,*V*, the value of I_{ei}^1 and I_{ei}^2 for $i \in N(t)$ to ensure no point on evasion path *e* is contained in the interior of more than one such line segment. Overlaps occur in five scenarios, as depicted in Figure 6. Scenario one occurs when both I_{ek}^1 and I_{ek}^2 intersection points fall between I_{ei}^1 and I_{ei}^2 where *k* > *i*. In this case, both I_{ek}^1 and I_{ek}^2 are set to 0 because all points on evasion path *e* in the segment $[(x_e, I_{ek}^2), (x_e, I_{ek}^1)]$ are also in the segment $[(x_e, I_{ei}^2), (x_e, I_{ei}^1)]$. The second scenario occurs when I_{ek}^1 falls between I_{ei}^1 and I_{ei}^2 and I_{ek}^1 falls below I_{ei}^2 where *k* > *i*. In this case, I_{ek}^1 is set to I_{ei}^2 so the points in $[(x_e, I_{ei}^2), (x_e, I_{ei}^1)]$ are now covered only once. The third scenario occurs when the exact opposite situation happens, I_{ek}^2 falls between I_{ei}^1 and I_{ei}^2 and I_{ek}^1 falls above I_{ei}^1. In this case, I_{ek}^2 is set to I_{ei}^1 by similar reasoning. The last scenario is a rare case where both I_{ek}^1 = I_{ei}^1 and I_{ek}^2 = I_{ei}^2. In this case, both I_{ek}^1 and I_{ek}^2 are set to 0. Lines 5 and 6 detail a two-dimensional matrix of intersected lengths for each node *i* and each evasion path *e*. Because of the algorithmic work done in Line 4, calculating intersected length for evasion path *e* is as simple as summing the difference $I_{ei}^1 - I_{ei}^2$ over $i \in N(0)$. Line 7 calculates the total intersected length by taking the minimum intersected length among all evasion paths.

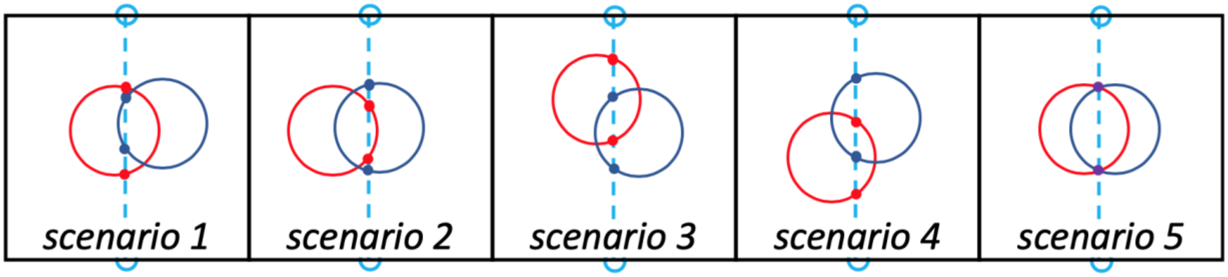


Figure 6: Five overlap scenarios found in the intersected length model. The red sensor corresponds with sensor i and the blue sensor corresponds with sensor k.

The k-coverage model, as shown in Figure 7, includes a binary detection matrix, a simple algorithm, and a formula for calculating coverage. Line 1 defines the binary detection matrix that holds a value of 1 when an evasion *e* is detected by node *i* and a value of 0 otherwise. In Line 2, values are assigned to I_{ei}^{0} for all *e* = 1…*V* and $i \in N(0)\{0\}$ by comparing x_i to x_e. If the horizontal distance from the center of node *i* to evasion path *e* is less than *r*, then a value of 1 is assigned, and otherwise, a value of 0 is assigned. Line 3 calculates the coverage for each evasion path *e* = 1…*V* by summing the number of nodes $i \in N(0)\{0\}$ for which $C_i(s) = 1$ and $I_{ei}^{0} = 1$. The network coverage $K_2(t_1, t_2, \dots, t_n; s)$ is then calculated as such value over all evasion paths *e* = 1…*V*.

1: I_{ei}^{0} = binary matrix indicating whether evasion path *e* = 1,…,*V* is covered by sensor $i \in N(0)\backslash\{0\}$
2: **For** *e* = 1: *V*
    For *i* = 1: *n*
        If abs$(x_i - x_e) < r$
            $I_{ei}^{0} = 1$;
        Else $I_{ei}^{0} = 0$
3: $K_2(t_1, t_2, \dots, t_n; s)$ = Min $\{\sum_{i=1}^{n} (I_{ei}^{0} * C_i(s)) : e = 1,\dots,V\}$

Figure 7: Representation of the k-coverage along an evasion path model

The minimum distance metric, which can be calculated using the algorithm in Figure 8, uses the distance I_{ei}^{0} between node $i \in N(0)\{0\}$ and evasion path *e* = 1…*V*. Line 2 calculates the horizontal distance between each node and each evasion path. Line 3 details the formula for calculating coverage using the minimum distance metric. First, the distance for each evasion path *e* = 1…*V* is obtained as the minimum value of I_{ei}^{0} away $i \in N(0)\{0\}$ with $C_i(s) = 1$. The network coverage $K_3(t_1, t_2, \dots, t_n; s)$ is reported as the maximum such distance over all evasion paths.

1: $I^0_{ei} =$ matrix of distance between node i and evasion path e for $i = 0,\dots,n$ and $e = 1,\dots,V$
2: **For** $e = 1{:}\ V$
  For $i = 1{:}\ n$
    $I^0_{ei} = \text{abs}(x_i - x_e)$
3: $K_3(t_1, t_2, \dots, t_n; s) = \text{Max}\{\ \text{Min}\ \{I^0_{ei} : i = 1, \dots, n \text{ and } C_i(s) = 1\} : e = 1, \dots, V\}$

Figure 8: Representation of the minimum distance model

2.3 WSN Failure Simulation

To determine the significance of our new total intersected length metric, we created a MATLAB program to simulate the failure process of a WSN. In each iteration of the simulation, we generate a network topology and a time to failure of each sensor node. Within each iteration, we track the three coverage metrics over time. A detailed summary of the simulation follows.

In each iteration, the simulation first generates an initial network topology defined by $N(0)$ and $E(0)$. The number of sensor nodes $n = |N(0)\backslash\{0\}|$ is input to the simulation. The location (x_i, y_i) of each sensor node $i \in N(0)\backslash\{0\}$ is randomly and independently selected within B according to a given probability distribution, with the sink node placed directly in the center of B for simplicity. Given $(x_i, y_i), i \in N(0)$, the edges $E(0)$ are generated as described in Section 2.1.

After generating the network topology, the time to failure t_i of each sensor node $i \in N(0)\backslash\{0\}$ is independently generated according to an Exponential(λ_i) distribution with rate

$$\lambda_i = 2\,\frac{\max_{j \in N(0)\backslash\{0\}}\{\text{distance from sensor } j \text{ to the sink}\}}{\text{distance from sensor } i \text{ to the sink}}. \quad (1)$$

The specific form of Equation (1) was chosen to impose that sensor nodes closer to the sink, which are more heavily relied upon to transmit data, tend to fail more quickly than sensor nodes that are farther away. The value 2 is not of significance because it simply scales the time-to-failure by a constant.

Next, V evasion paths are generated running vertically within B. The intersection locations (x_e, b) and $(x_e, 0)$ of each evasion path $e \in V$ are determined by

$$x_e = e * \frac{a}{V+1}. \quad (2)$$

After the network is fully initialized, the times to failure are sorted from smallest to largest as described in section 2.1. This sorted list provides a timeline through which the simulation uses to model passing time. Beginning with the smallest t_i and pausing at each t_i, the model progresses through time, updating the network and measuring and recording each coverage metric as it goes. The process described above is repeated S times, and the output metrics are averaged across the S iterations.

2.4 Illustration on Notional Network

Figure 9 represents an example in which B = [0,4] × [0,1] includes N (0) = {0, 1, 2, 3, 4, 5}.

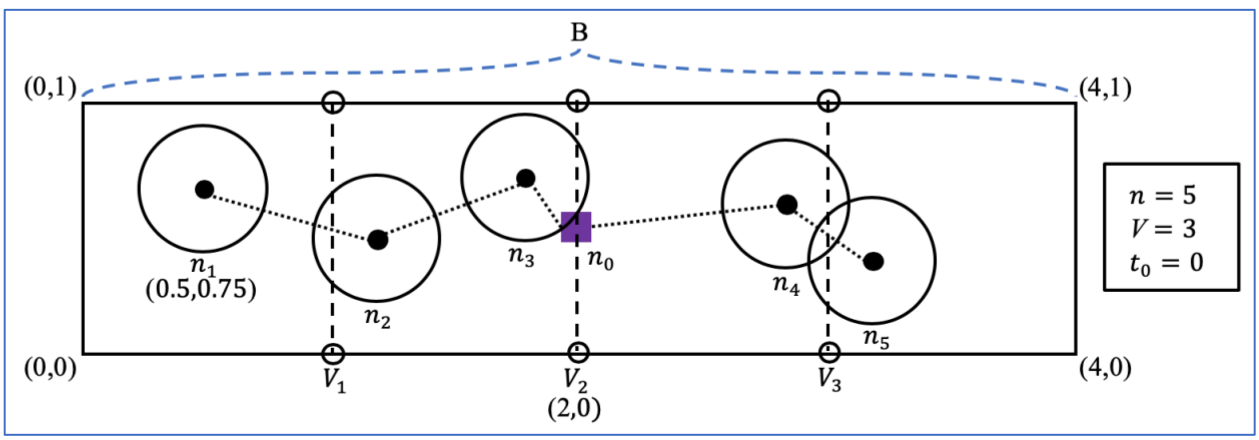


Figure 9: An example WSN at time 0. Five sensors are distributed across border region B and three evasion paths run through the region

The network's state at time $t_0 \equiv 0$ is replicated in Figure 9, and the coverage metrics are illustrated in Figure 10. The total intersected length metric (shown in green) finds the distance between all intersection points at each evasion path and then records the smallest value, 0.2. The minimum distance metric (shown in blue) finds the distance between each evasion path and its closest node and records the largest value, 0.2. The k-coverage metric (shown in red) counts the number of sensors that each evasion path intersects and records the smallest value, 1.

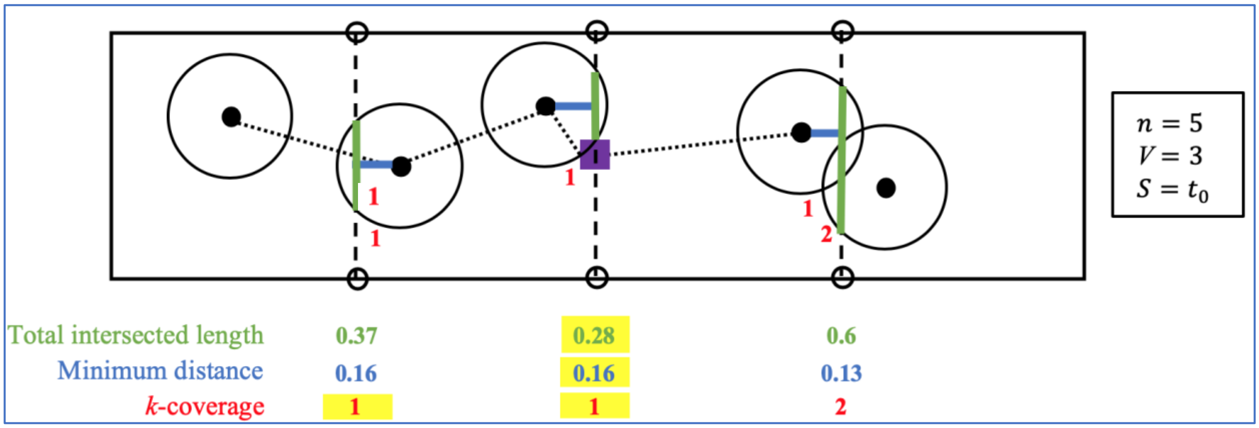


Figure 10: An example WSN is measured using each coverage metric at time 0.

Suppose node n_5 fails first, i.e., $t_1 = t_5$. Each metric is recalculated as demonstrated in Figure 11. The total intersected length value and the k-coverage values for evasion path V_3 decrease while the minimum distance value remains the same. Although values change along evasion path V_3, the coverage value does not change for any of the three metrics because the changed values do not exist on the least covered evasion path.

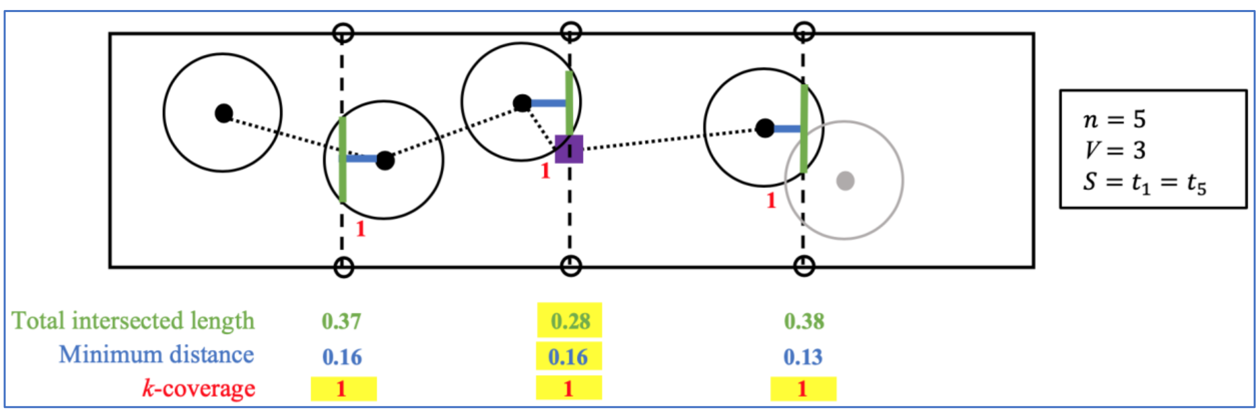


Figure 11: An example WSN is measured using each coverage metric at time 1.

Suppose node n_2 fails next. Because d $[(x_1, y_1), (x_3, y_3)] > r$, n_1 becomes disconnected as well, see Figure 12. Each coverage metric is updated and, because evasion path V_1 is not detected by any live sensors, records a value of 0.

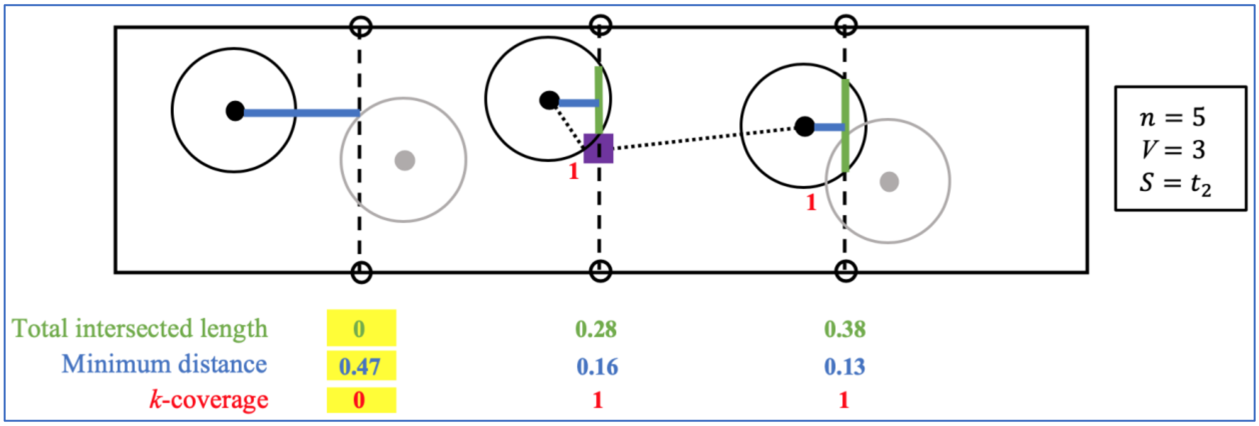


Figure 12: An example WSN is measured using each coverage metric at time 2.

3. Numerical Results

Using numerical experimentation, we compared and contrasted the three metrics under a number of different network designs (i.e., as specified by the probability distribution governing the placement of nodes in B). We specifically investigated how each metric evolves for each design as the network degrades. Section 3.1 describes the experimental setup followed by a discussion of the results for each metric in Section 3.2 – 3.4 and comparison of each metric's results in Section 3.5. Section 3.6 describes how the results for each metric differ when the assumed distance-dependent failure rate is eliminated.

3.1 Experimental Settings

Initializing our experiment WSN, we set $n = 200$ within a border region $B = [0,10] \times [0,1]$. On the y-axis, sensors were uniformly distributed, and, on the x-axis, sensors were distributed according to a truncated normal distribution with mean equal to five and standard deviation equal to σ. Sensors were arbitrarily given $r = 0.5$, and communication distance was set to $r_c = 2.0$. To have enough evasion paths to span B and for the distance between each evasion path to be less than r, we set $V = 50$.

Next, we set up several WSN *designs* to investivate how nodes should be located in the border region. We hypothesized that all of the metrics would prefer for sensors to be more concentratred near the center of the region, e.g., by using a relatively small value of σ in the truncated normal distribution. The rational for this hypothesis is because the sink node is placed in the center of the border region; thus, sensors near the edges of the region depend on many sensors to communicate with the sink node while sensors near the middle of the region depend on far fewer sensors to connect with the sink node. This characteristic makes the sensors near the center of the region more valuable to the overall network integrity than the sensors closer to the edges [10].

To determine the most effective distribution, networks were simulated with a variety of σ values for the truncated normal distribution. We refer to the networks generated under $\sigma = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,$ and ∞ as "design SDσ," where SD∞ corresponds to using a uniform distribution to locate sensors in the x-direction. In Figure 13, a few of the truncated normal distributions are compared to the uniform distribution.

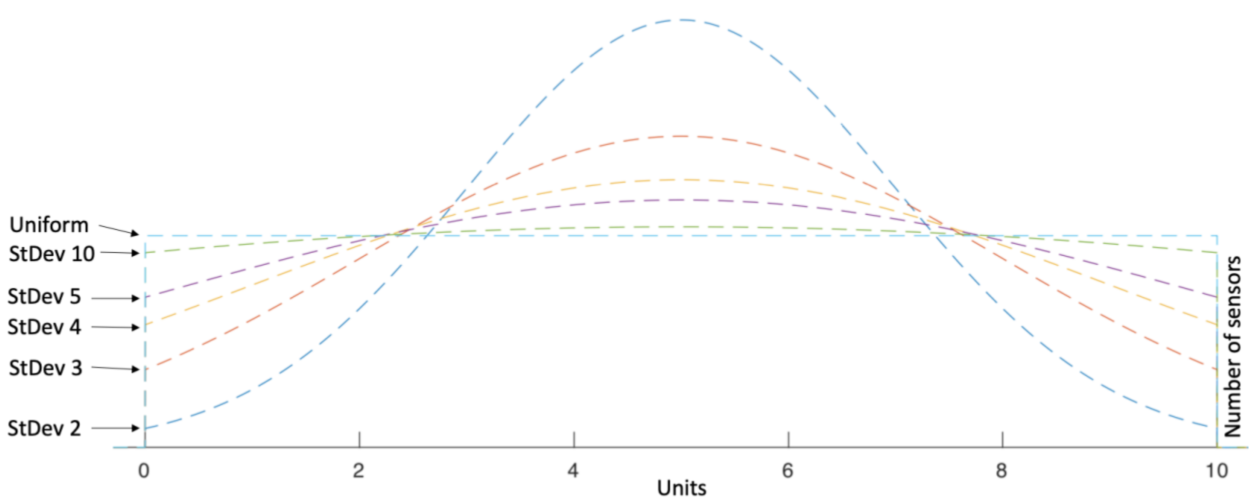


Figure 13: Truncated normal distributions with different σ values compared to the uniform distribution

We simulated 2000 replications of each design, outputting all three metrics defined in section 2.2. We used n = 200 sensors in the intial network. We chose n = 200 because we noticed that the networks with greater than 200 sensors offered few observable differences while costing significantly greater computation times.

3.2 Total Intersected Length Metric Results

The results from measuring coverage using the first metric, total intersected length, were as follows. For SD∞, the metric, on average, measured close to full coverage (1 unit after normalizing by dividing by the length of an evasion path) for a short period and then dropped off steadily until it nears zero when around 40 sensors remained, see Figure 14. The initial coverage measure is understood to be the result of a well spread-out network that effectively covered every section of every evasion path. As time passed and a number of sensors failed, either fewer live sensors covered the least-covered evasion path or sensors along critical pathways failed, leaving nodes live but unconnected, or effectively dead. This pattern continued until one evasion path was completely uncovered by live and connected sensors. At this point, the metric measured coverage to be zero.

When the network sensors were distributed according to SD2, the metric initially measured the network to be less than fully covered (0.7 units) and then dropped off quickly from there, with the value nearing zero when around 20 sensors remained, see Figure 14. The initial value of around 0.7 is understood to be the result of far fewer sensors being distributed to the edges of the network, with the majority of sensors accumulating near the center of the border region. As time passed, sensors failed, causing the overall coverage to decrease, but, because there were more sensors around the sink to keep far nodes connected, the network degraded slower and maintained effective coverage longer, with the value reaching zero with 15 fewer sensors. This tradeoff between initial coverage effectiveness and network longevity was observed for each of the three metrics.

In our pursuit of the most desirable distribution according to each metric, we found that no one distribution was overall better than all the others. Instead, we observed a wide range of tradeoffs. Design SDσ with smaller values of σ provided worse initial coverage but maintained coverage for longer, and, as the σ was increased, the initial network coverage grew closer to 1 and the network remained covered for longer. For total intersected length, maximum initial coverage was met as early as SD4, see Figure 14. This means the most significant tradeoffs occur when the network standard deviation is four or less, as any distribution with a standard deviation greater than four measured as good as or less coverage than SD4.

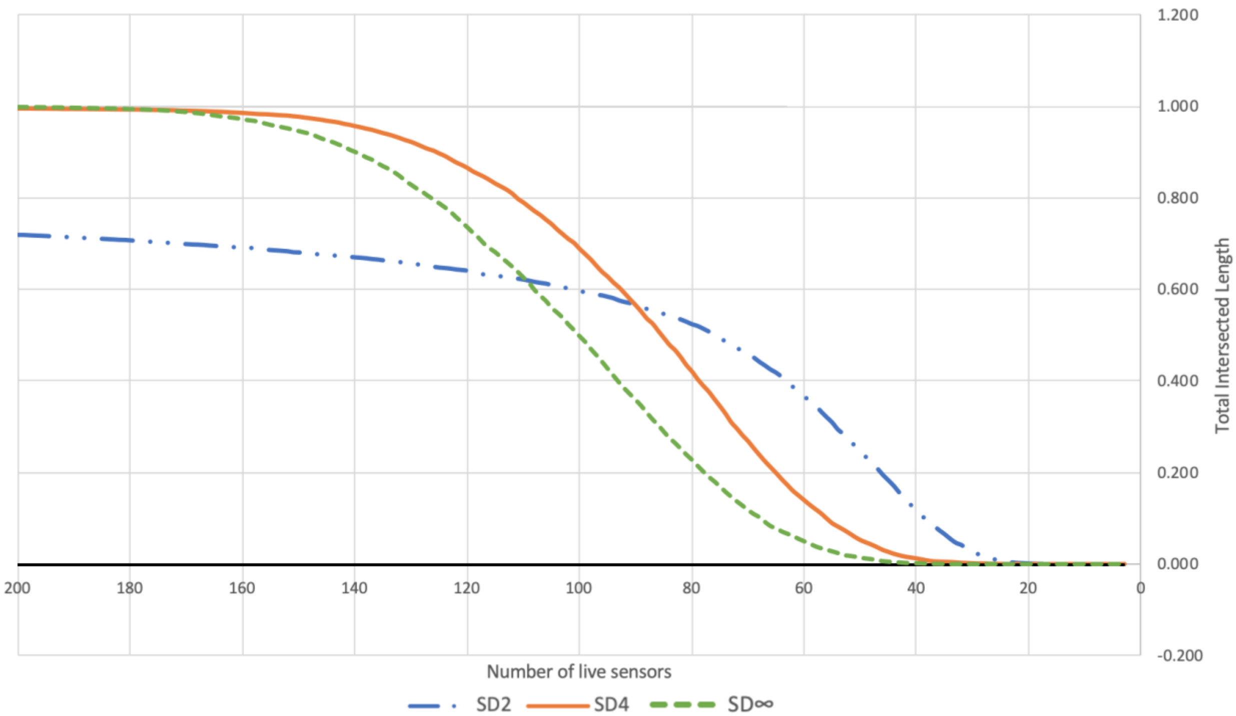


Figure 14: Total intersected length metric for three different sensor distributions: SD∞, SD4, and SD2

3.3 *K*-coverage Metric Results

The results from measuring coverage using the *k*-coverage metric were as follows. When sensors were distributed according to SD∞, *k*-coverage measured initially around 12 units and then dropped steadily until around 50 sensors remained, see Figure 15. The initial *k*-coverage is understood to be the result of, on average, at least 12 sensors covering each evasion path. As time passed, *k*-coverage decreased until the metric reached a measure of zero units of coverage, when around 50 sensors remained live.

For design SD2, the metric measured initial *k*-coverage to be significantly less, with at least 2.5 sensors covering each evasion path. Unlike *k*-coverage when the network was distributed according to a uniform distribution, the *k*-coverage at SD2 changed little when the network lost its first 100 sensors. At around 100 sensors remaining, *k*-coverage began to decrease quickly until *k*-coverage was measured to be zero when around 20 sensors remained, see Figure 15. This behavior is believed to be the result of few sensors initially covering the outer edges of the network, resulting in low initial coverage values, and the majority of failing sensors being concentrated toward the middle of the network where sensor failure had little impact of the least covered evasion paths near the network edges. Once the sensors near the network edges did begin to fail, the network became vulnerable quickly. Interestingly, as seen with the total intersected length metric, the network lasted for longer using the SD2 distribution than it did when using the uniform distribution.

As standard deviation was increased, the initial coverage increased, but not until SD8 did the coverage metric initially measure the same, 12 units of coverage, as with SD∞, see Figure 15. Any increase in σ beyond SD8 showed little variation in coverage across time. The distributions between SD2 and SD8 showed significant tradeoffs between initial *k*-coverage effectiveness and lifetime of the network.

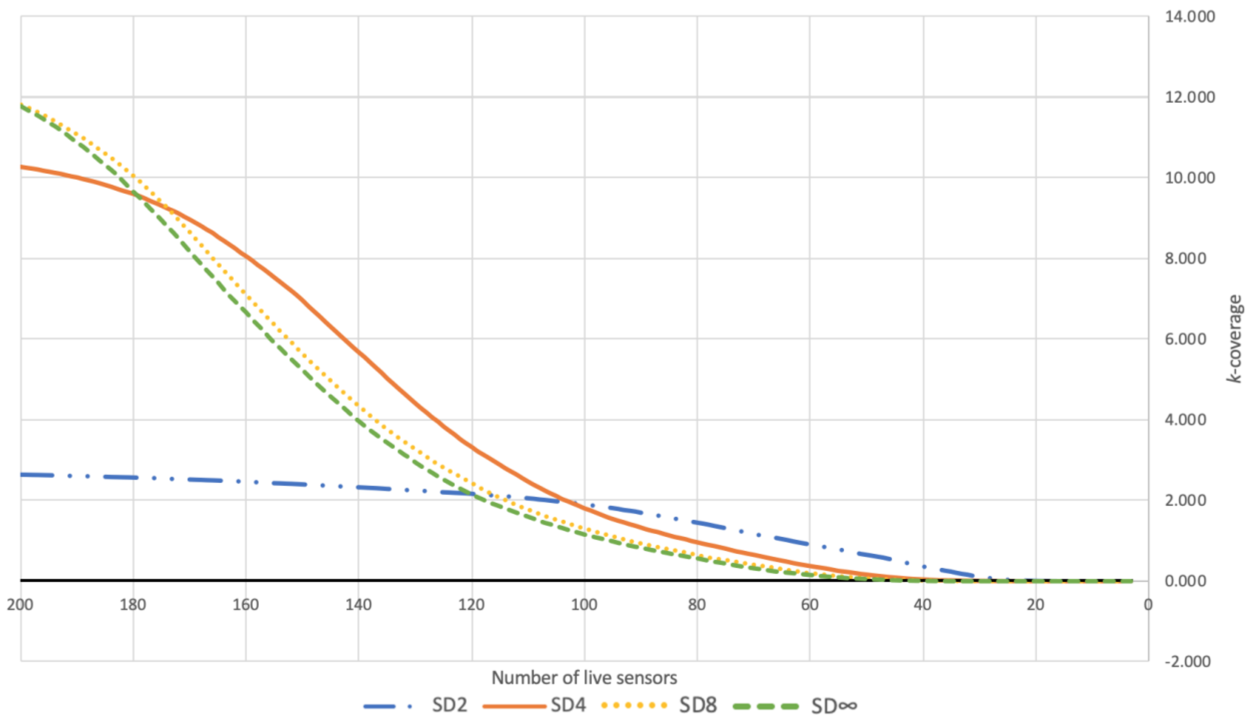


Figure 15: K-coverage metric for three different sensor distributions: SD∞, SD4, and SD2

3.4 Minimum Distance Metric Results

Unlike the other metrics, the minimum distance starts small and increases over time. Initially, for a network with SD∞, the minimum distance was measured at around 0.1 units of coverage, see Figure 16. This measurement increased slowly while the first 120 sensors failed and then increased dramatically while the remaining 80 sensors failed. The shape of Figure 16 is believed to be the result of a well-distributed initial network that degraded across time until around 80 sensors remained. At this point, the network was no longer considered effectively covered. This is because each node has a maximum sensing distance of 0.5 units, and when the minimum distance between any evasion path and its closest sensor surpasses 0.5 units, then at least one evasion path is left completely uncovered. The measurements beyond 80 sensors remaining increase quickly and can be ignored because they are not relevant to this research.

When the network was set to SD2, the initial coverage value according to the minimum distance metric was around 0.25 units of coverage, see Figure 16. After remaining almost constant for a period, the network coverage metric began to increase until when around 50 sensors remained. At this point, the coverage passed 0.5 units, rendering the network insufficiently covered. As with the other coverage metrics, the initial coverage value is worse when and the network remains covered for longer when the network is set to SD2 than when the network is distributed according to SD∞.

As with the total intersected length metric, coverage, when the network was set to SD4, had the same initial value (~0.1) as the network with SD∞, see Figure 16. Any distribution with SDσ larger than 4 is going to fail quickly and provide no additional initial coverage. This means that the significant tradeoffs within the network occur for designs SDσ with σ < 4. More coverage curves can be found in the appendix.

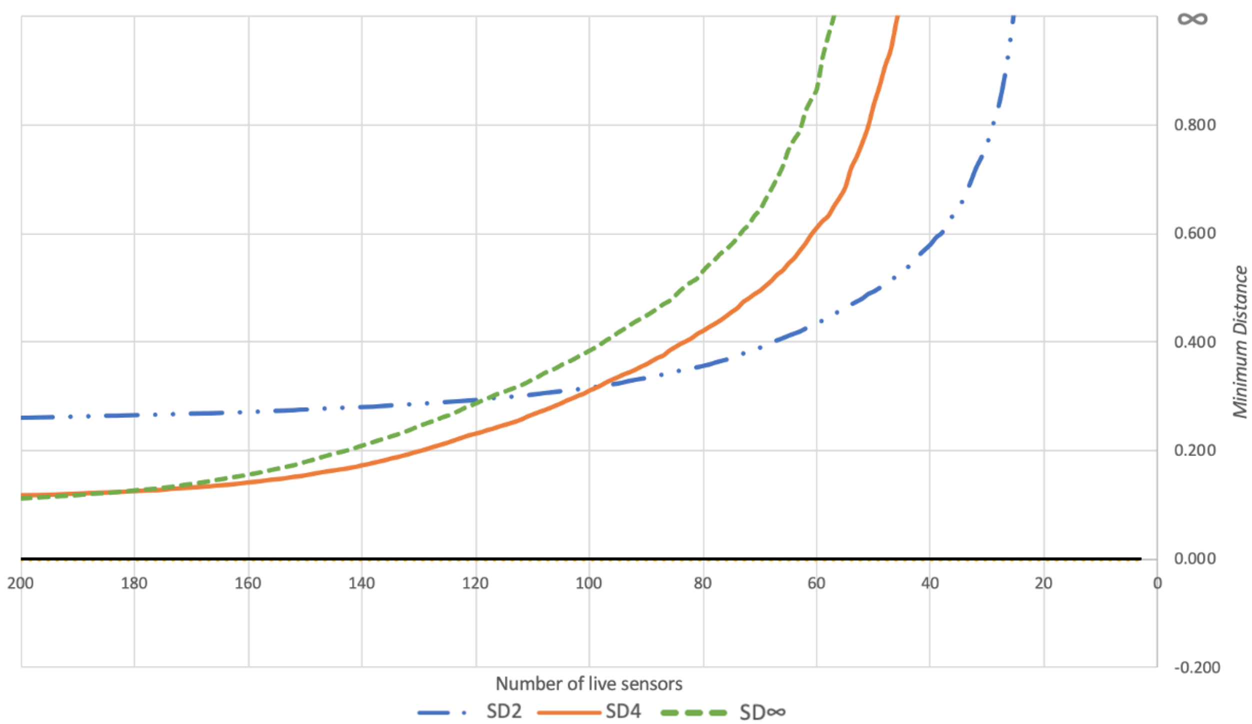


Figure 16: Minimum distance metric for three different sensor distributions: uniform, normal truncated with a standard deviation of four, and normal truncated with a standard deviation of two.

3.5 Comparison of the Three Metrics

It is important to note that all three metrics were output for the same simulated networks. The observed differences are not a product of how the network degraded, but rather a difference in how each metric interpreted the amount of coverage in the network

As mentioned previously, we observed tradeoffs between initial coverage and network longevity for each coverage metric. For each metric, the SD∞ network showed a better coverage than that for SD2, but by the end of the network's useful life, the network set at SD2 showed a greater coverage measure than that for the uniform for each metric. Between these two end values exists a point where the coverage is measured the same for both uniform and SD2, see Figure 17. This point is where a center-heavy network becomes preferable to a more spread out network. For application requiring coverage over long time periods, a more center-heavy network might be preferable, but the exact point at which the preferred network type switches depends on how coverage is quantified. For the total intersected length metric, this point occurred when around 110 sensors remained while for the other two metrics this point occurred when around 120 sensors remained. This is interpreted as neither an advantage nor a disadvantage of the total intersected length metric, but rather as an interesting finding worth noting. While the conventional coverage metrics may agree about the exact point at where the distribution type tradeoff occurs, the total intersected length metric and possibly other metrics disagree. This finding could be useful information for some problems involving optimizing a WSN to maximize coverage.

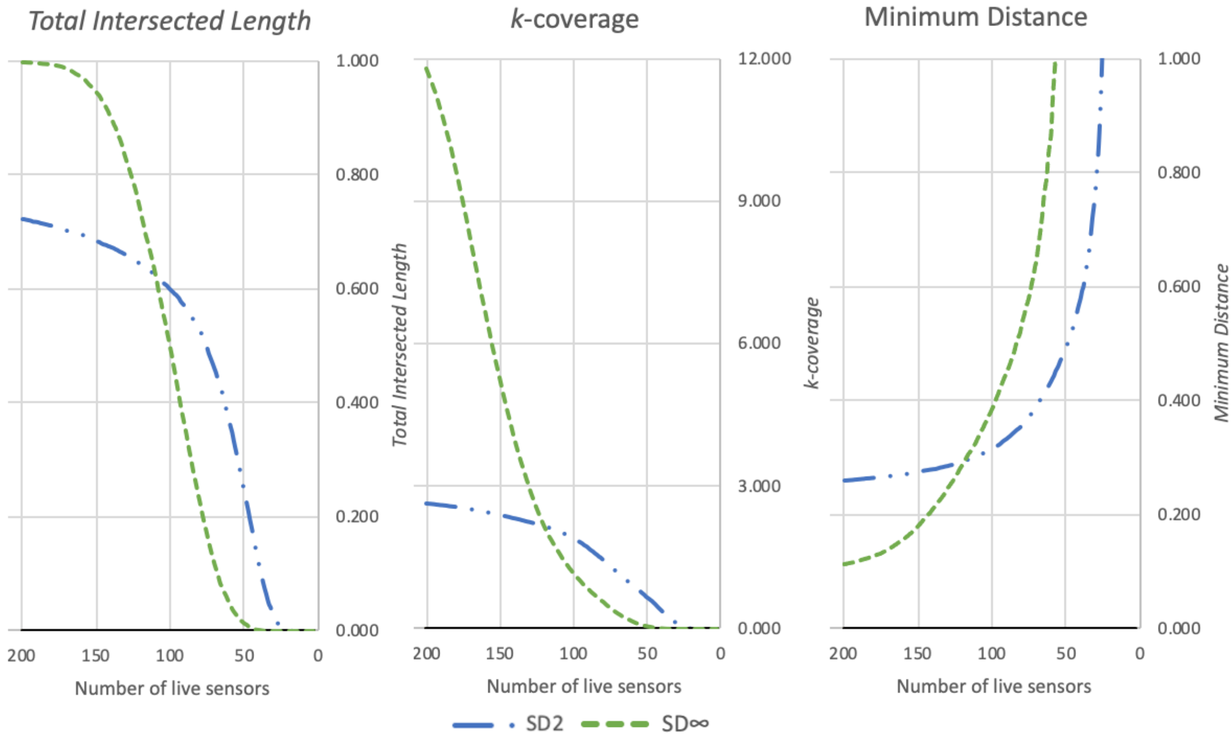


Figure 17: The tradeoff point between a network distributed according to a uniform distribution and SD2 for each coverage metric, total intersected length, k-coverage, and minimum distance, respectively

Another interesting difference among the metrics is how much worse the initial coverage was measured to be when the network was set to SD2 as compared to when the network was set to SD∞. The total intersected length metric interpreted the least difference between initial coverages, with initial coverage being 72% of initial coverage for the uniform when set to SD2. Somewhat similarly, initial coverage is 231% of initial coverage for SD2 when set to uniform, according to the minimum distance metric. The *k*-coverage metric interpreted the widest variation in initial coverage, with initial coverage being 22.5% of initial coverage for SD∞ when set to SD2. Because we know from the total intersected length measurement that an average of at least 72% of each evasion path is covered when the network is set to SD2, the dramatic difference between the initial coverage when set to SD∞ and the initial coverage when set to SD2 according to the *k*-coverage metric could be considered unrealistic. Because of this, we consider the total intersected length's method for measuring initial coverage to be an advantage, at least over the *k*-coverage metric.

Surprisingly, the average point at which each coverage metric interpreted the network as insufficiently covered, that is the average value where at least one evasion path is completely uncovered, also varied. For the total intersected length metric, the network reached insufficient coverage when around 40 sensors remained for the uniform distribution and when around 25 sensors remained for SD2. According to the *k*-coverage metric, the network reached insufficient coverage when around 30 sensors remained for the uniform distribution and when around 20 sensors remained for SD2. Lastly, the minimum distance metric interpreted that the network reached insufficient coverage when around 85 sensors remained for the uniform distribution and when around 50 sensors remained for SD2. Since we do not know when, on average, the networks actually became insufficiently covered, we cannot determine which metric most accurately captured this. It seems likely that the discrepancy in values in a consequence of using averages, which can be easily skewed by a few outliers. The most conservative measurements were found by the minimum distance metric. One possible explanation for this is that sometimes when a network failed, the next closest sensor was further from the evasion path than r, 0.5 units. If this explanation was, in fact, true, then the metric is likely too conservative with its measurement. Falling in the middle of the two traditional coverage metrics, the total intersected length metric offers a less conservative measurement than the minimum distance metric but a more conservative measurement than the *k*-coverage metric, possibly an advantage of the total intersected length metric depending on the application.

Beyond significant points found on the metric curves, the metrics showed varying rates of degradation across time for the SD∞ design, see Figure 18. The total intersected length metric curve showed the majority of the network's coverage degradation happens between when 140 and 60 sensors remained. The k-coverage metric curve interpreted the network's coverage as decaying earlier, with the majority of degradation taking place between when 200 and 100 sensors remained.

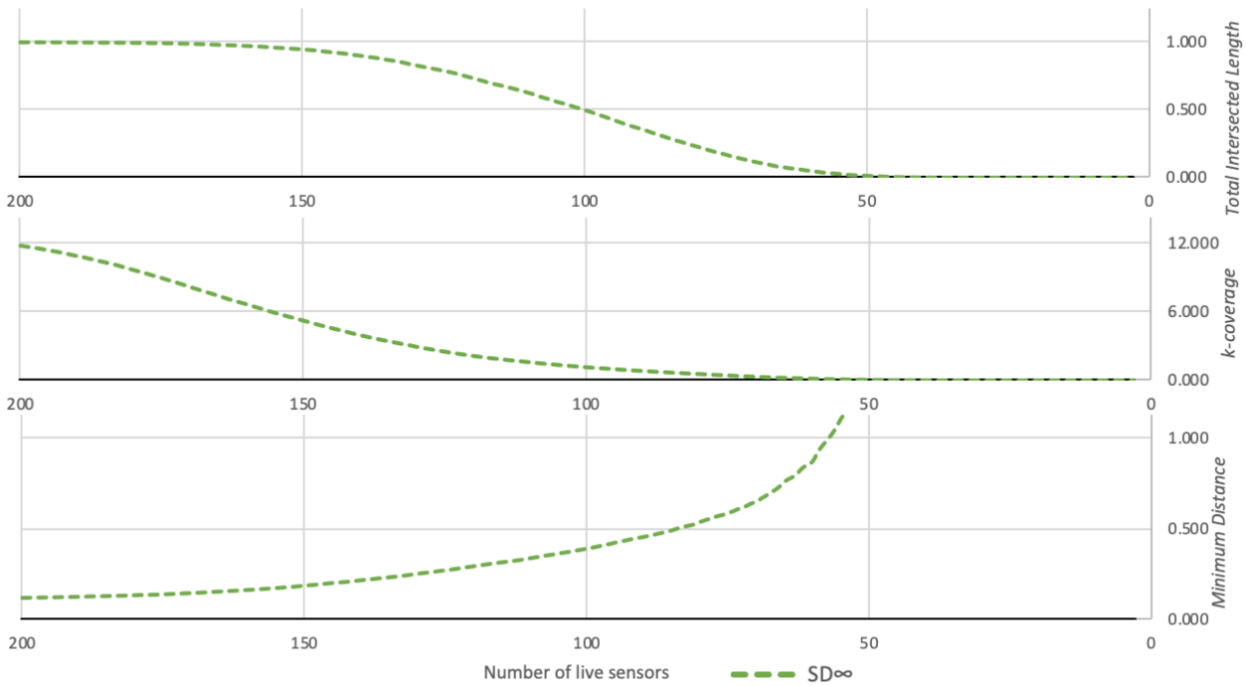


Figure 18: The degradation rates for a uniformly distributed network according to each coverage metric, total intersected length, k-coverage, and minimum distance, respectively

The minimum distance metric curve showed a similar degradation patten as the total intersected length metric curve, with the majority of its relevant degradation happening when the last 160 sensors failed. Of the three metrics, the total intersected length metric shows the most coverage for the longest period. This tendency to possibly overestimate the coverage of the network could be considered a disadvantage of the metric.

Possibly the biggest disadvantage of the total intersected length metric we found is its computation time. Because the *k*-coverage and minimum distance metrics can be quickly updated without having to run time-consuming loops, updating the measures across time is relatively quick. The total intersected length metric, however, requires a triple loop calculation to account for sensor overlap every time a sensor fails. As shown in Figure 19, there is minimal difference between computation time for the networks when the number of sensors in the network is small, but as the number of sensors grows, the computation time for the total intersected length metric dramatically outpaces that for either the *k*-coverage or the minimum distance metrics.

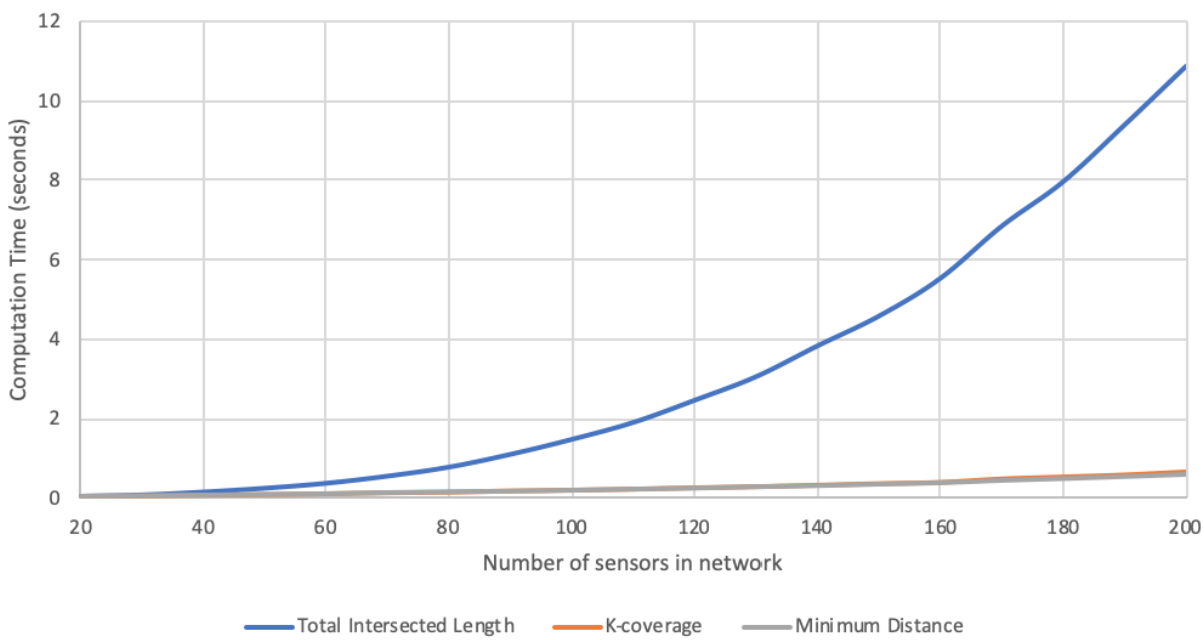


Figure 19: Computation time by the number of sensors in the network for each coverage metric

3.6 Sensitivity to the Assumed Distance-Dependent Failure Rates

Understanding that our linear distance-dependent failure rate possibly represents an exaggerated version of the energy hole created in WSNs, we simulated the model separately under the assumption that no distance-dependent failure rate exists, see Appendix 4-6. These results suggest that results relating to network design seem to depend on the assumption that sensors closer to the sink fail faster. In reality, failure rates likely fall between these two extremes; thus, the true results of our metrics likely exist between the results in Appendix 1-3 and those in Appendix 4-6.

4. Conclusions

Through the results of this research, we can conclude that the total intersected length metric used for measuring network coverage for border security WSN applications has both advantages and disadvantages when compared to traditional coverage metrics. The new metric was found to be advantageous for measuring the initial condition of a network and for determining when a network is no longer sufficiently covered while it was found to be disadvantageous for conservatively capturing network degradation and for producing quick computation times. Each of these findings depends largely on the specific application in which the metrics are used.

Future research in the border security application will now have a broader selection of coverage metrics to choose from and, because of our analysis, researchers can see some of the potential tradeoffs of using any of the three metrics. We think this work will allow for more complete and helpful research in the optimization of WSNs in the border security application.

Future work on this topic could include verifying actual network fail time, determining additional tradeoffs among the metrics, and speeding up computation time for the total intersected length metric. Further analysis could be done using the same MATLAB program we created to measure how many sensors remained when each network failed. These data could be aggregated to form a fail time distribution along with which the metrics could be compared to determine which metric is the most accurate. The computation time for the total intersected length metric was minimized in this work, but further research into improving this algorithm's efficiency could help significantly improve the utility of this metric.

5. References

[1] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292-2330, 2008.

[2] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys (CSUR),* vol. 43, no. 4, pp. 1-53, 2011.

[3] I. F. W. S. Akyildiz, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks,* vol. 38, no. 4, pp. 393-422, 2002.

[4] P. Rawat, K. D. Singh, H. Chaouchi and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of supercomputing,* vol. 68, no. 1, pp. 1-48, 2014.

[5] R. M. Curry and J. C. Smith, "A survey of optimization algorithms for wireless sensor network lifetime maximization," *Computers & Industrial Engineering,* vol. 101, pp. 145-166, 2016.

[6] J. Chen and X. Koutsoukos, "Survey on coverage problems in wireless ad hoc sensor networks," *IEEE SouthEastCon,* vol. 2007, pp. 22-25, 2007.

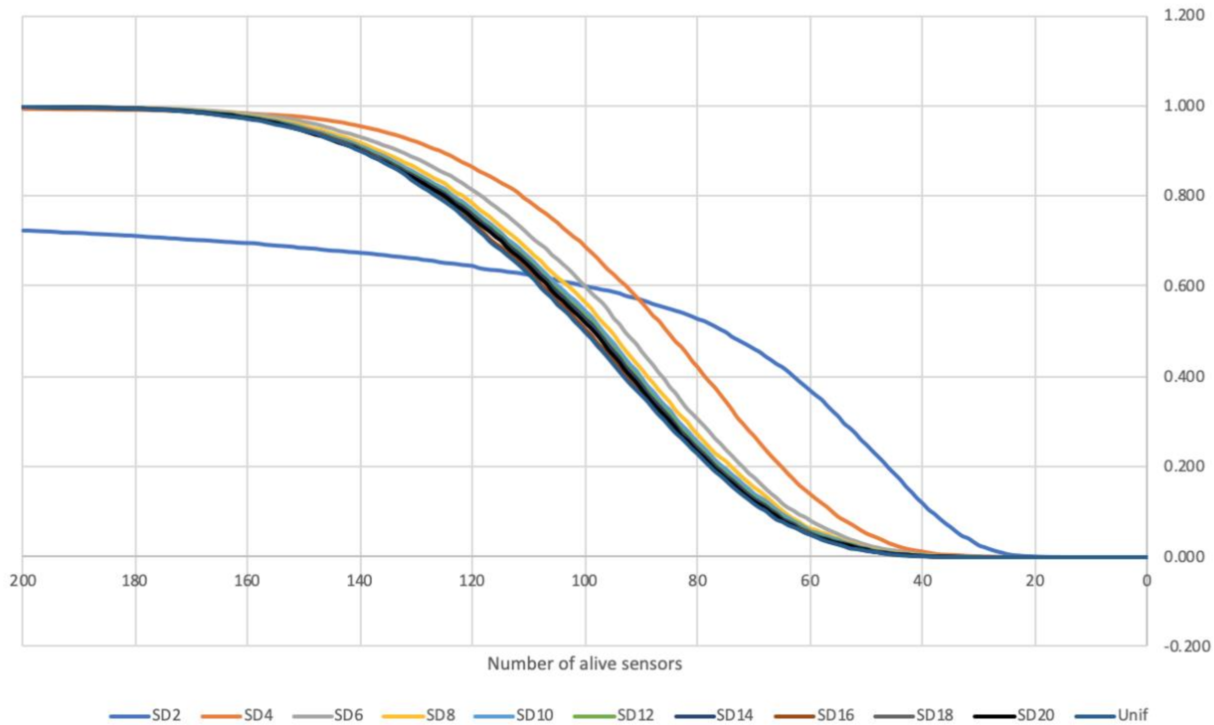
[7] H. M. Ammari, The art of wireless sensor networks, New York: Springer, 2014.

[8] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," *Proceedings IEEE INFOCOM 2001,* vol. 3, pp. 1380-1387, 2001.

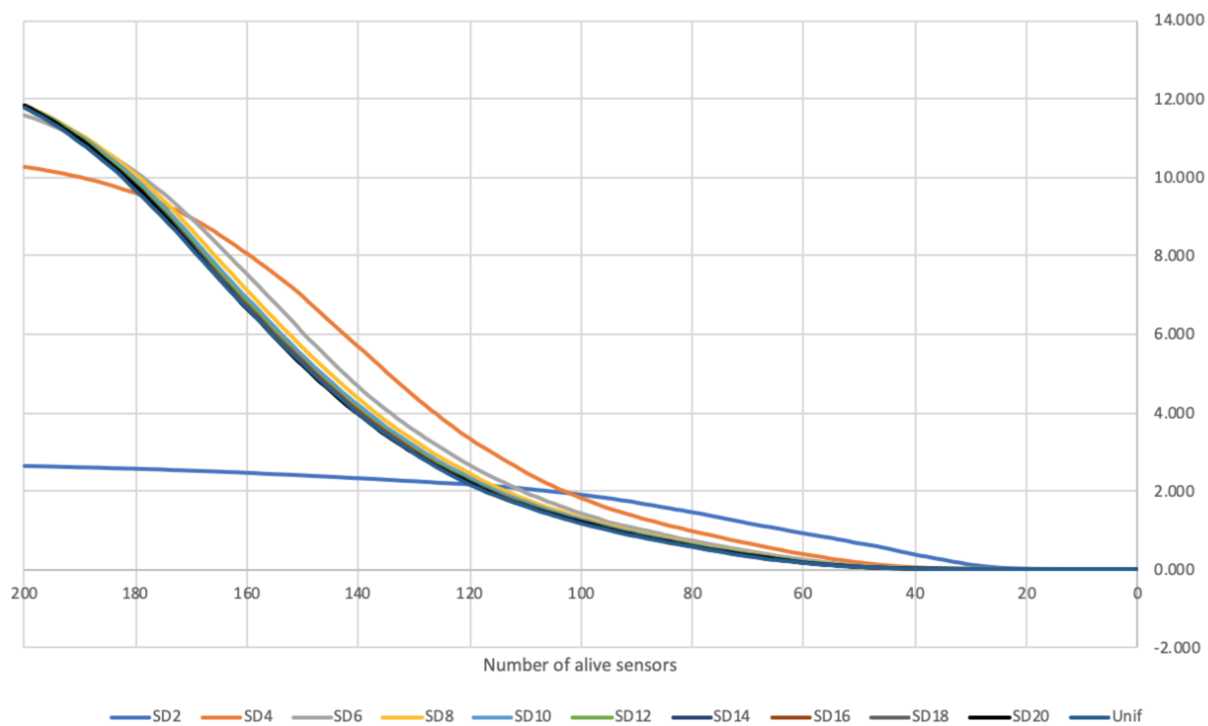
[9] A. Saipulla, C. Westphal, B. Liu and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks," *IEEE INFOCOM 2009,* pp. 127-135, 2009.

[10] J. Li and P. Mohapatra, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks," *Pervasive and mobile Computing,* vol. 3, no. 3, pp. 233-254, 2007.

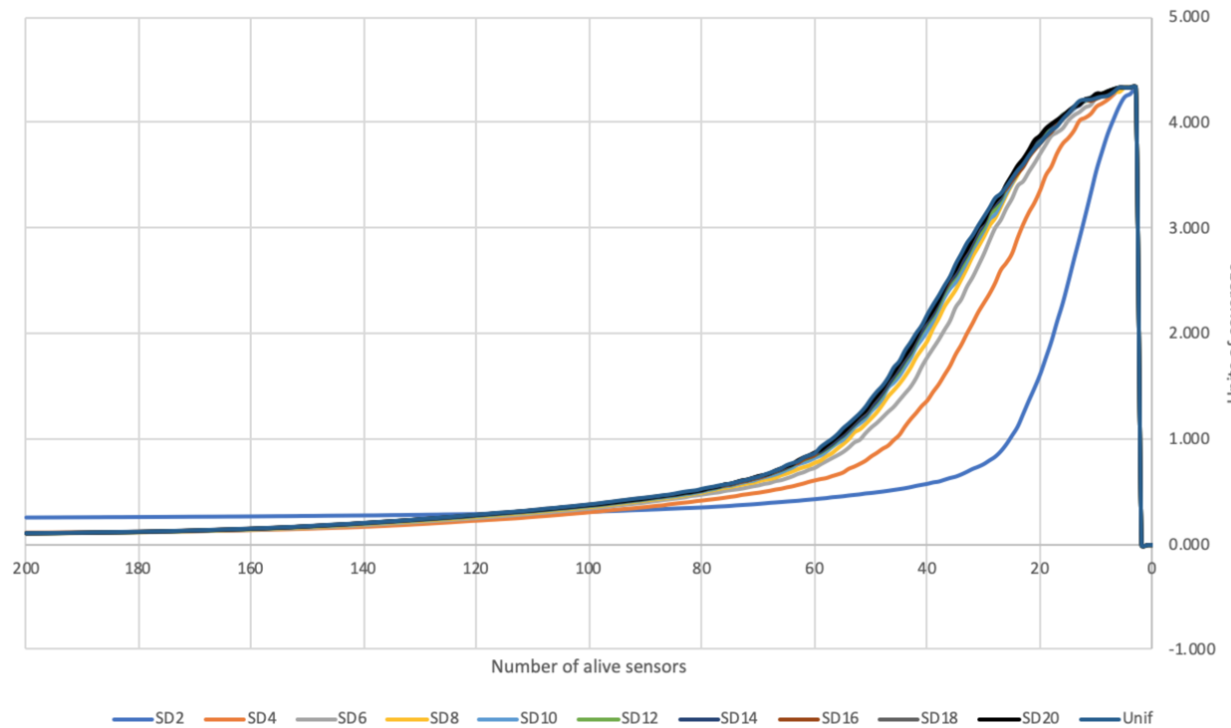
6. Appendix



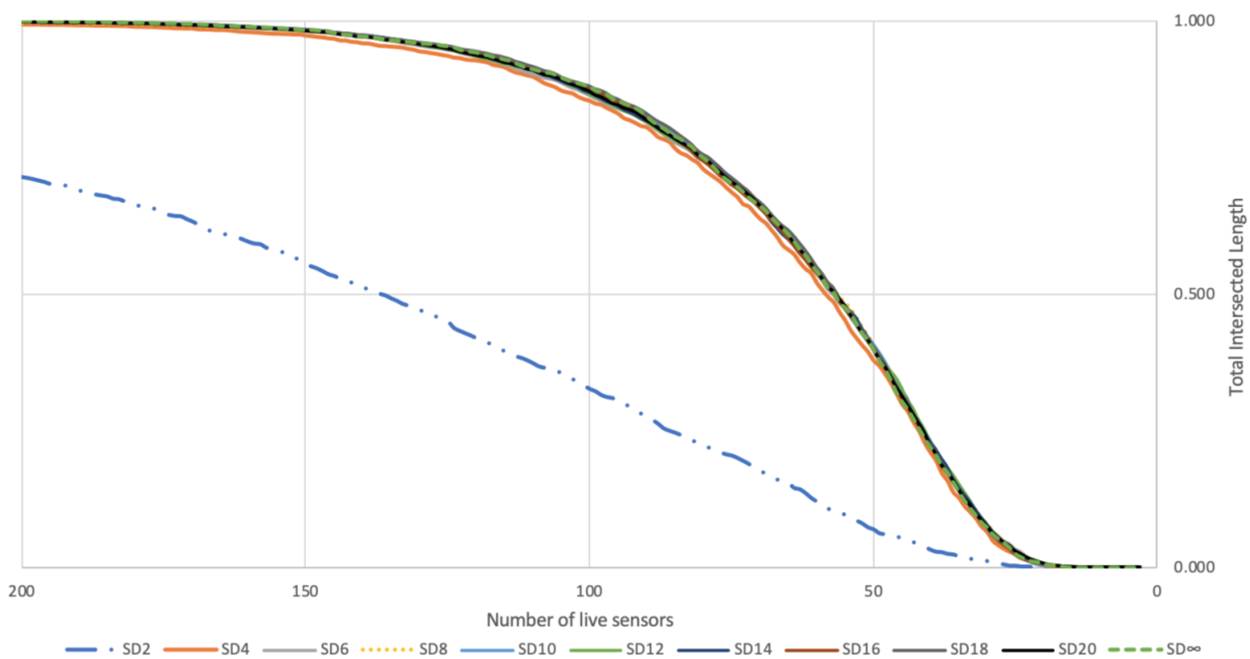
Appendix 1: Network degradation from 200 to 0 sensors with variable network distributions as measured by the total intersected length metric



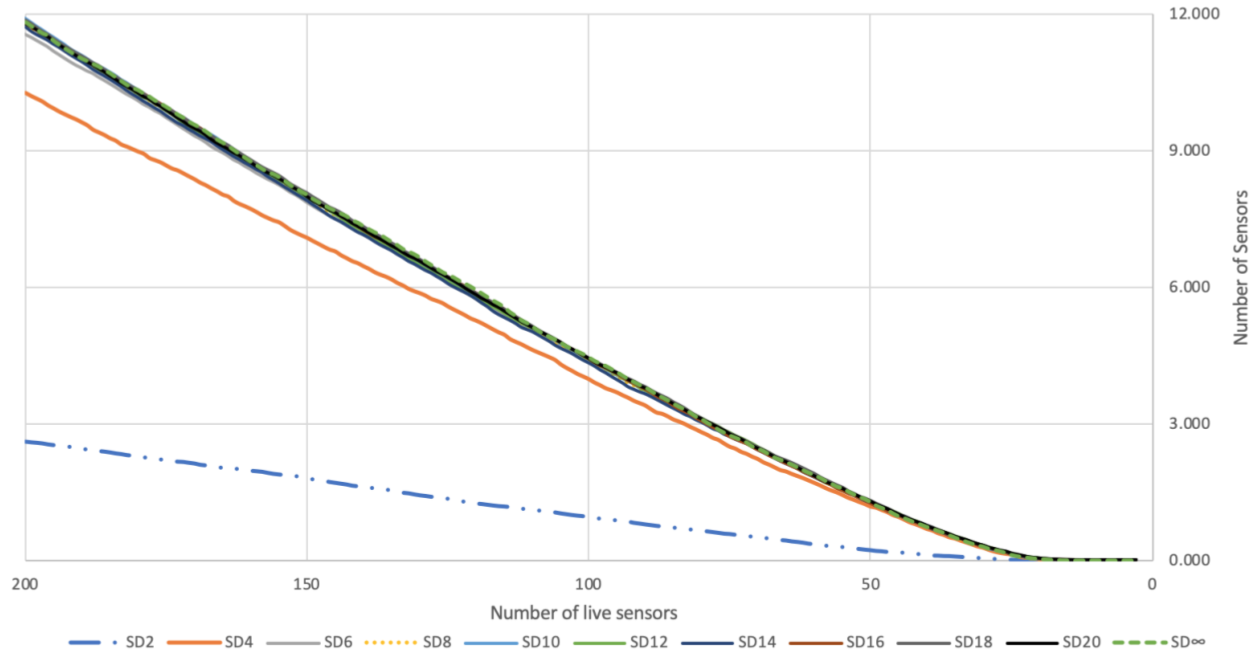
Appendix 2: Network degradation from 200 to 0 sensors with variable network distributions as measured by the k-coverage metric



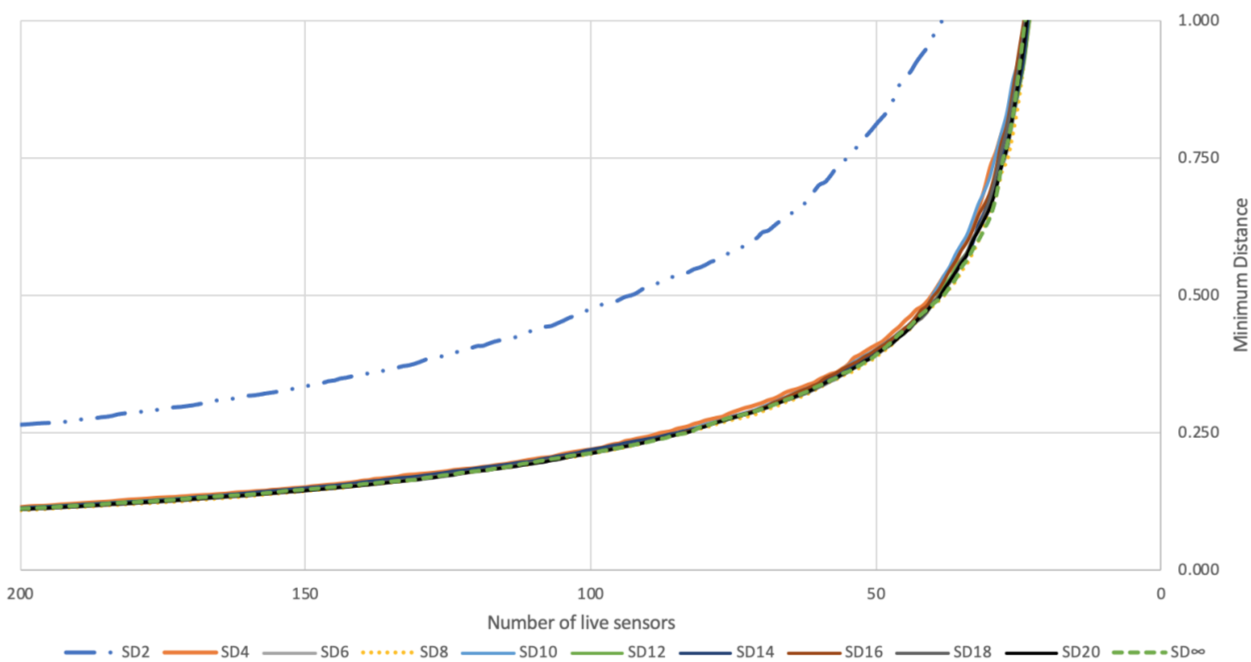
Appendix 3: Network degradation from 200 to 0 sensors with variable network distributions as measured by the minimum distance metric



Appendix 4: Network degradation from 200 to 0 sensors when the sensors have no distance-dependent fail rate, as measured by the total intersected length metric



Appendix 5: Network degradation from 200 to 0 sensors when the sensors have no distance-dependent fail rate, as measured by the k-coverage metric



Appendix 6: Network degradation from 200 to 0 sensors when the sensors have no distance-dependent fail rate, as measured by the minimum distance metric

Contact Dr. Kelly Sullivan at ksulliv@uark.edu
Contact K. Karlton Haney at kevinkarlton@gmail.com