TerrainLOS: An Outdoor Propagation Model for Realistic Sensor Network Simulation

Sam Mansfield
Department of Computer Engineering
University of California
Santa Cruz, CA, USA
Email: smansfie@ucsc.edu

Kerry Veenstra
Department of Computer Engineering
University of California
Santa Cruz, CA, USA
Email: veenstra@soe.ucsc.edu

Katia Obraczka
Department of Computer Engineering
University of California
Santa Cruz, CA, USA
Email: katia@soe.ucsc.edu

***Abstract*—We present TerrainLOS, an outdoor propagation model that uses Digital Elevation Models to determine whether two nodes can communicate. We have implemented TerrainLOS in the sensor network simulator COOJA and used it to evaluate how the roughness of terrain, which we classify using Average Cumulative Visibility, affects the number of edges, connectedness, latency, and power of a network. We compare the difference in results when using TerrainLOS and a simpler propagation model to show how the performance of outdoor simulation is greatly affected by a model that takes terrain into account.**

I. INTRODUCTION

The advent of the Internet of Things, has renewed interest in remote sensing applications, such as monitoring air pollution or detecting forest fires. These applications rely on many small low-power, resource constrained devices scattered in an area, that infrequently send information to a gateway node for processing in the cloud.

Sensor networks have attracted considerable attention from the network research community and have been the subject of a vast body of work. Network simulation platforms have been particularly useful in the development, testing, and evaluation of sensor networks and their protocols. However, studies have questioned the fidelity of simulation platforms [1] [2] [3]. One of the main culprits is the channel propagation modeling used by network simulators in general, and sensor network simulators in particular, which tend to be overly simplistic and do not accurately simulate reality. For instance, MobiCom and MobiHoc publications from 1995 to 2003 were reviewed in [4], which reported that the use of overly simplistic propagation models outweighed accurate models. The same study also noted that routing protocols perform differently depending on the propagation model used. Consequently, even if a study is only comparing two different algorithms, the propagation model itself can determine the outcome.

In this paper we propose a new terrain-aware propagation model, that we call TerrainLOS (Terrain Line Of Sight). Our model aims to strike a balance between complexity and accuracy in outdoor simulation. We use Digital Elevation Models (DEMs) [5] to determine whether wireless nodes can communicate. TerrainLOS is computationally simple and can be calculated at simulation time. We aim its use to more accurately simulate outdoor sensor networks and serve as a complement to testbeds.

We use TerrainLOS to evaluate different classes of terrain, which we classify using an extension of the metric used by Veenstra and Obraczka, Average Cumulative Visibility [6] (ACV). We analyze the different classes of terrain based on the number of edges formed in the network, as well as the largest partition in the network. We also analyze the latency and power when running two different routing protocols, RPL [7] and ORPL [8].

Our results show that terrain affects simulation. We find that terrain quickly disconnects the network. We also find that RPL has less variability in power consumption across classes of terrain, whereas ORPL has less variability in latency.

II. DESIGN

TerrainLOS was designed to balance complexity and accuracy when simulating sensor network deployments in outdoor environments. The goal was to provide a model that offered: (1) better accuracy than a Unit Disk Graph Model (UDGM), the more common name for the Simple Earth Model, described by Newport et al. [4], where the only criteria to determine whether two nodes can communicate is the distance between them, and (2) less complexity than Ray Tracing, where every ray is calculated from source to destination.

As described in more detail in Section II-A below, TerrainLOS uses Wang et al.'s viewshed algorithm [9] to determine whether two nodes can communicate. This algorithm was chosen because it can calculate viewsheds with low complexity.

Additionally, our model uses Digital Elevation Models (DEMs) provided by the Shuttle Radar Topology Mission (SRTM) [10] which provide DEMs for most of the world.

Currently, TerrainLOS does not incorporate path loss, nor transmission-induced errors, and assumes a circular transmission range. We plan in future work to add more realism to the model.

A. Wang Viewshed Algorithm

Wang et al.'s viewshed algorithm [9] generates a viewshed, a geographic map of points that are visible from a given viewpoint. Each node has its own viewshed, which it uses to determine whether communication is possible.

To determine the visibility of a point from a given viewpoint, Wang et al.'s viewshed algorithm computes the height that the point must have for it to be visible. If the actual height of the point is equal to or exceeds this amount, then the point is deemed visible.

The algorithm uses dynamic programming to compute these required heights. Each point's required height is computed from the actual height of the point, the actual height of the viewpoint, and the required heights of any points that are immediately adjacent to the point and are closer to the viewpoint. Given these relationships, required-height computation is seeded with the actual height of the viewpoint and the actual heights of the viewpoint's eight immediate neighbors. Then required-height computation proceeds outwards from the viewpoint in concentric, square rings. The algorithm terminates when it has determined the required heights of the farthest points. Then the visibility of any point can be determined by comparing its required height with its actual height.

To ensure proper operation of the algorithm, the required height of a point is never allowed to be "underground." Since the required height of a point is used by each of its immediate neighbors to determine their own visibility, a point's required height never can be less than the height of the terrain below the point. The algorithm computes the required height of each location once, meaning that the algorithm's complexity depends on the number of points on the map. After all required heights are computed, the visibility determination for any point is a constant time calculation.

B. Height Maps

We chose to use height maps created by the SRTM [10], a joint venture between the National Aeronautics and Space Administration (NASA) and the National Geospace Intelligence Agency (NGA). The goal of the SRTM is to topographically map 80% of the earth's surface every 1-arc second (approximately every $30m$ or $100ft$) The accuracy of the elevations is $16m$, with 90% confidence (taken from the SRTM Mission Statement).

We are limited by the 1-arc second resolution, of the SRTM height maps, but since this study focuses on the effects of terrain on simulation, as opposed to the accuracy of simulation versus deployment, we chose not to seek higher resolution height maps.

III. IMPLEMENTATION

We implemented TerrainLOS in COOJA [11] a simulator for the Contiki [12] operating system. We chose COOJA because it is used frequently for sensor network applications.

Contiki is a lightweight operating system for memory constrained devices, such as sensor nodes. Contiki's features are preemptive multi-threading, dynamic loading of programs at runtime, and a small memory footprint [12].

A. TerrainLOS Radio Medium

COOJA is written in Java and implements propagation models as *Radio Media*. A specific Radio Medium is called when a node wants to communicate with another node. The Medium determines whether the communication is possible.

We implemented our TerrainLOS propagation model as a COOJA Radio Medium. TerrainLOS extends the Unit Disk Graph Model (UDGM), which was previously implemented in COOJA by Fredrik Österlind. The UDGM Radio Medium creates a directed graph of nodes within transmission distance of each other. There are additional features, such as transmission and reception percentages, which determine the percentage of packets that will not be corrupted on transmission and reception, respectively. We chose to keep the percentages at 100%, because we did not incorporate data loss into our model.

We extended UDGM by only adding nodes to the directed graph if they have line of sight with each other, determined based on Wang et al.'s viewshed algorithm using SRTM DEMs.

IV. EXPERIMENTAL METHODOLOGY

The goal of our experiments is to validate our claim that terrain must be taken into account when evaluating sensor network outdoor deployments as it can significantly affect experimental results and lead to incorrect conclusions. To this end, we carry out a comparative study in which we run simulation experiments using the proposed TerrainLOS model and compare the results against the same experiments that are instead run with UDGM.

In our study, we conduct two different sets of experiments. The first set examines the effect of terrain on network connectivity and uses *number of network edges* and *network connectedness* as metrics. The second set of experiments investigates how terrain impacts network protocol performance. In particular, we study two well-known sensor network routing protocols, namely RPL [7] and ORPL [8] and used *latency* and *power consumption* as evaluation metrics.

A. Classification of Terrain

In order to use a variety of SRTM tiles in our experiments, it was important to be able to quantify their terrain features, e.g., how "flat" or "hilly" they are.

We use Average Cumulative Visibility (ACV) based on Wang et al.'s viewshed algorithm as used in [6]. The ACV

Table I
PARAMETERS ADJUSTED FOR EXPERIMENTATION

Independent Parameters	How Parameters are Adjusted
Node Layout	Uses one random uniform distribution of nodes.
Number of Nodes	100
Area	9,000,000 m^2
Density 5	R = 415 m
Density 9	R = 508 m
Density 16	R = 677 m
ACV 5%-100%	Adjusted by using different 100 by 100 tiles of SRTM data.
ContikiMAC Wakeup Interval	500 ms
ORPL EDC ω	0.5

is defined as the average of the area visible from all locations on the map. To calculate the ACV we iterate through each location in the DEM, calculate the percentage of the DEM that is visible at that location based on its viewshed, and average over all the locations.

B. Node Placement and Density

In our experiments, we use 100 nodes randomly distributed in a 100 arc-second by 100 arc-second area, which is equivalent to 100 by 100 digital elevations of a DEM (approximately 3,000 m by 3,000 m).

We then vary node density which is given by the formula: $\mu(R) = (N * \pi * R^2)/A$ [15], where R is the node's transmission range, N is the number of nodes, and A is the area of the region being simulated. Since N and A are constants in our simulations, we adjust R to vary node density. Table I summarizes COOJA's parameters that are relevant to our experiments and the values we use in our simulations.

C. Network Connectivity Metrics

When COOJA runs the TerrainLOS Radio Medium, it creates a directed graph of all the nodes that are visible to each other. We then computed two different metrics for the graph, namely: (1) the average degree of each node and (2) the largest partition containing the sink node, which we fixed to be node 1. While the average degree of a node is given by the number of directly connected neighbors the node has, the largest partition containing the sink node expresses how connected the network is.

D. Latency and Power Consumption Metrics

For the set of experiments in which we study how terrain affects network protocol performance, we used data delivery latency and power consumption to compare two sensor network routing protocols, RPL [7] and ORPL [8]. We chose RPL as it is an IETF standard for sensor network routing. ORPL is an evolution of RPL that uses opportunistic routing. The ORPL authors evaluated ORPL against RPL [8] and one of the goals of this set of experiments is to try to reproduce their results using TerrainLOS. Below we provide a brief description of RPL and ORPL.

RPL computes a directed acyclic graph with the sink node as the root. Each node has only one parent. To deliver a packet, a node sends upwards to the first shared ancestor, and then the ancestor sends downwards until it reaches the desired node. Since we are running a collect application, traffic is only routed upwards.

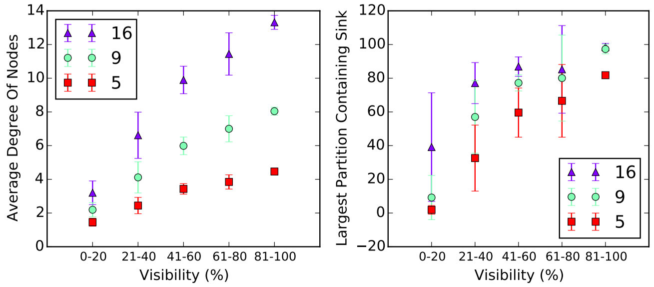
ORPL also forms a directed acyclic graph with the sink node as the root. But, ORPL is opportunistic, a node can have multiple parents. A packet is sent the same way as in RPL, but because each node has multiple parents, data can reach a destination in less time, because if opportunism exists, bottlenecks through one node will be reduced. ORPL can also reduce power consumption, because if opportunism exists nodes will spend less time transmitting, because there are more nodes willing to forward and acknowledge the data, reducing the amount of time a node has to transmit.

Duquennoy et al. [8], the authors of ORPL, evaluated their protocol in the Indriya testbed [16], which is an indoor 3-D testbed. They showed that ORPL was able to achieve lower latencies and lower power usage than RPL in both collect and any-to-any applications.

We use the same simulation setup described in [8] which is available from https://github.com/simonduq/orpl. The sink node, which in our case is node 1, only listens and does not perform any cycling, i.e., its radio is always on. The remaining nodes, send a 64B packet over UDP, at a random time within four minutes. Each node uses ContikiMAC [17] with a radio duty cycle of $500ms$, which means they wakeup every $500ms$ to listen for packets and send continuously for $500ms$ or until the transmission is acknowledged. ORPL uses EDC (Expected Duty Cycles) [18] to form its forwarding graph, and we fix the ω parameter, the cost of forwarding, to 0.5, because this was found to be a good balance for ORPL and RPL [8].

Data delivery latency is calculated as the average time it takes for a packet to arrive at the sink after it was sent. The average is calculated over all packets received by the sink. If a packet is sent and never arrives, it is not factored into the latency calculations or if a node is not part of the partition containing the sink, this is also not factored into the latency calculations.

Power consumption is measured using the average Radio Duty Cycle (RDC), i.e., the percentage of time that a node's radio is on. RDC is a widely used metric for power consumption in sensor networks, because sensor nodes' power consumption tends to be dominated by their radio [8].



(a) Average Degree

(b) Largest Partition

Figure 1. Figure 1(a) depicts the average degree for nodes, plotted against the ACV, for the densities 5, 9, and 16. Figure 1(b) depicts the largest partition containing the sink node, plotted against the ACV, for the densities 5, 9, and 16.

V. RESULTS

This section presents the results using the evaluation metrics described in Section IV. Note that one of our baselines, the Unit Disk Graph Model (UDGM) corresponds to an ACV of 100%. Each data point in the graphs represents the average of all runs in the specified ACV range, and a minimum of five data points are collected per range. The error bars represent one standard deviation of the mean.

A. *Network Connectivity*

We focused our edge analysis on three densities, 5, 9, and 16. These were chosen because they depict a nice spread of the data, as seen in Figure 1(a) and Figure 1(b).

1) Average Degree Vs. ACV: The average degree appears to be linearly proportional to ACV. This is not surprising. As ACV decreases, fewer nodes are visible from each node, which is the same as saying the number of edges between the nodes decreases.

If we instead did not use TerrainLOS, and used UDGM to simulate an outdoor scenario on a terrain at 50% ACV, the simulation would assume double the degree of what TerrainLOS would use. For example if the density is set to 16, on top of a 50% ACV terrain, UDGM would be using on average 8 more edges per node to deliver packets, which could lead to different performance results.

2) Largest Partition Vs. ACV: The largest partition containing the sink has a more interesting shape then the average degree, which can be seen in Figure 1(b). It appears to be linear with respect to ACV from 100% to around 40%, where there is a steep drop off. When the density is 16, there is a much larger spread of data points compared to the lower densities.

Another interesting thing to note is when the density is 5, even with an ACV of 100%, the network is never fully connected, and instead maxes out around 80 nodes. This is not unexpected, and was stated by Kleinrock and Silvester [19] in previous work.

If we instead used UDGM, we would assume, at densities of 9 and 16, that the network is fully connected, but we can see that this is not the case, and with ACV as high as 80%, with a density of 16, the network is already disconnected.

B. *Latency in RPL vs. ORPL*

In Figures 2(a) and 2(b) we can see the latency when we run RPL and ORPL with densities of 5 and 16. We only show the densities of 5 and 16 because we find that it shows an accurate spread of results.

RPL: We will first examine RPL. At high ACV, latency is low, but as ACV decreases to 40% for density 5, and 30% ACV for density 16, latency is at its highest, and then drops to very close to zero as ACV approaches 0%.

To explain this phenomenon we can look at our earlier graph analysis. As the ACV decreases the partition containing the sink node, drops off slowly until it reaches a tipping point, around 40% ACV for density 5, and around 30% ACV for density 16, as we saw in Figure 1(b). The average degree at each node also decreases, but more steadily, as we saw in Figure 1(a). Right before the respective densities rapidly decrease their partition size, they face the worst partition size to average degree ratio. Each node in RPL only has one parent, at high ratios of partition size to average degree, there are fewer paths to the sink node, and it is more likely to have congestion. We can verify this by examining the Packet Delivery Ratio in Figures 2(c) and 2(d), which we can see does dip at the same point.

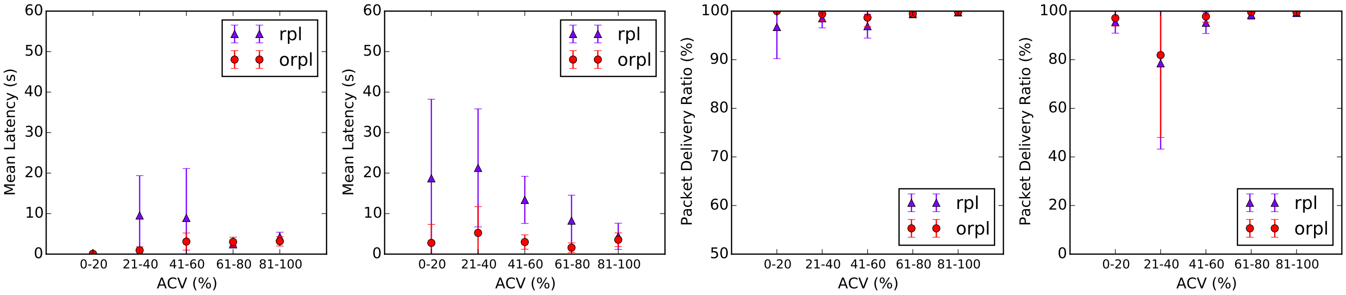
At 100% ACV, which would correspond to using UDGM, instead of TerrainLOS, we have a comparable, but higher latency than Duquennoy et. al [8], but as ACV drops, and especially as ACV approaches 40% for density 5, and 30% for density 16, latency goes as high as $50s$ and $30s$ respectively. This is a factor of 10 difference between latencies using UDGM and TerrainLOS.

ORPL: ORPL maintains a steady latency across ACV. ORPL is able to take advantage of the opportunism, and even in the large partition and low average degree case, which increased RPL's latency, ORPL's latency remains steady, even though the Packet Delivery Ratio does also dip at the same point as RPL.

We would expect ORPL to achieve lower latencies at higher densities, as Duquennoy et al. point out, ORPL does not perform as well at lower densities, because there is less opportunism for ORPL to take advantage of [8]. At 100% ACV ORPL does have a slightly lower latency at density 16, then at density of 5, but as ACV decreases their latencies are comparable.

ORPL's latency is not as affected by ACV, as RPL, and simulations using UDGM, would achieve similar results across ACV, even without our model.

RPL vs. ORPL: We do see that ORPL has lower latency than RPL, even at lower densities, which is consistent with Duquennoy et al. [8], but the ratio of their differences varies



(a) Latency: Density 5

(b) Latency: Density 16

(c) PDR: Density 5

(d) PDR: Density 16

Figure 2. Figure 2(a) and 2(b) compare RPL's and ORPL's latency vs. ACV at densities 5 and 16. Figure 2(c) and 2(d) compares RPL's and ORPL's Packet Delivery Ratio vs. ACV at densities 5 and 16.

a lot, depending on the ACV. If we use our model on top of a terrain with 30% ACV, and a density of 16, ORPL would outperform RPL with a ratio over 10 to 1, but UDGM, would ignore the terrain effects, and instead report that there latencies are similar.

C. Power Consumption in RPL vs. ORPL

RPL: In our experiments, RPL's RDC (Radio Duty Cycle), which can be seen in Figures 3(a) and 3(b) stayed consistent across different ACV's, ranging from a minimum of 1% for both densities, to a maximum of 1.2% for density 5, and a maximum of 1.4% for a density of 16. The RDC can never be lower than 1%, because the sink node, always has its radio on, and because there are 100 nodes, this will always contribute 1% to the average RDC. The RDC is slightly higher at 100% ACV, because a larger portion of the nodes are sending and receiving data.

We see comparable RDC with Duquennoy et al., where they report a RDC of 1%, for RPL, using our same settings.

If we used UDGM, instead of TerrainLOS, RPL's RDC would be reported higher, for terrain with ACV less than 60%.

ORPL: ORPL's RDC is more varied than RPL's. For density 5, the RDC decreases slowly as ACV decreases, until 40%, and then rapidly deceases to 1%. For density 16, the RDC peaks around 30% ACV, and decreases in both directions from this point.

It is not a coincidence that these are the same ACVs where RPL had higher latency. ORPL has to compensate for its low latency by having a higher RDC to overcome the shortcomings of RPL.

For density 5, 40% ACV isn't the peak RDC, because 5 is such a low density, the added nodes sending and receiving as ACV increases to 100%, is enough to increase the Radio Duty Cycle above the RDC at 40% ACV, but we do see the rapid drop off as ACV decreases to 0%.

Our results do not match with Duquennoy et al. [8]. Their results showed ORPL having closer to half the RDC we report. If we look at the average number of hops in Figures 3(c) and 3(d), we can see that on average ORPL has more hops per transmission, which is consistent with what we are seeing for the RDC. Potential future work could be to investigate this further.

UDGM across terrain would misreport that ORPL always has a Radio Duty Cycle of 1.4%, but this varies over terrain. The RDC varies from close to 1% at 20% ACV at density 5, to 1.8% ACV at density 16.

RPL vs. ORPL: As mentioned in the previous section our analysis shows that ORPL has the same or worse RDC than RPL. This is not consistent with Duquennoy et al. [8].

If we use UDGM, instead of TerrainLOS, we would report that ORPL performs 1.167 ($1.4/1.2 = 1.167$) times worse than RPL, but this changes over ACV, going as small as 0 times worse, 20% ACV at density 5, to approximately 1.5 ($1.8/1.2 = 1.5$) times worse, 30% ACV at density 16.

VI. CONCLUSION

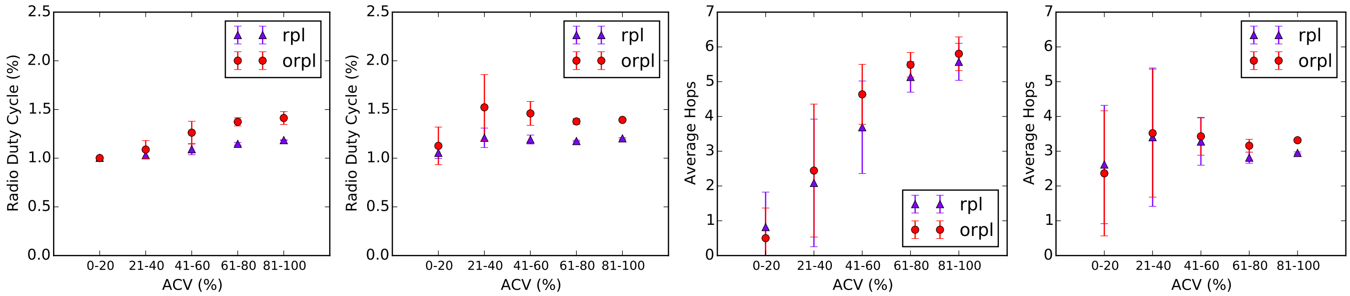
In this paper, we introduced a new propagation model called TerrainLOS that uses Digital Elevation Models (DEMs) to model outdoor propagation. We implemented TerrainLOS in the COOJA sensor network simulator and showed how it affects network topology as well as performance of core network protocols, such as routing. We ran our experiments in a variety of terrain samples, which we classify using the Average Cumulative Visibility (ACV) metric. Directions for future work include incorporating more realistic signal propagation models into TerrainLOS and cross-validating its results against real deployments.

ACKNOWLEDGMENT

This work has been partially supported by NSF under project CNS 1321151.

REFERENCES

[1] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee, "On credibility of simulation studies of telecommunication networks," *Communications Magazine, IEEE*, vol. 40, no. 1, pp. 132–139, 2002.



(a) RDC: Density 5
(b) RDC: Density 16
(c) Hops: Density 5
(d) Hops: Density 16

Figure 3. Figures 3(a) and 3(b) compare RPL's and ORPL's average Radio Duty Cycle vs. ACV at densities 5 and 16. Figures 3(c) and 3(d) compare RPL's and ORPL's average hops vs. ACV at densities 5 and 16.

[2] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.-c. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of detail in wireless network simulation," in *Proceedings of the SCS multiconference on distributed simulation*, 2001, pp. 3–11.

[3] M. D. Yarvis, S. W. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring, "Real-world experiences with an interactive ad hoc sensor network," in *Parallel Processing Workshops, 2002. Proceedings. International Conference on.* IEEE, 2002, pp. 143–151.

[4] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," *Simulation*, vol. 83, no. 9, pp. 643–661, 2007.

[5] K. Chang, *Introduction to geographic information systems*, 6th ed. McGraw-Hll, 2012.

[6] K. Veenstra and K. Obraczka, "Guiding sensor-node deployment over 2.5 D terrain," in *Communications (ICC), 2015 IEEE International Conference on.* IEEE, 2015, pp. 6719–6725.

[7] T. Tsvetkov, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," *Sensor NodesOperation, Network and Application (SN)*, vol. 59, p. 2, 2011.

[8] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the tree bloom: scalable opportunistic routing with ORPL," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems.* ACM, 2013, p. 2.

[9] J. Wang, G. J. Robinson, and K. White, "Generating viewsheds without using sightlines," *Photogrammetric engineering and remote sensing*, vol. 66, no. 1, pp. 87–90, 2000.

[10] E. Rodriguez, C. S. Morris, J. E. Belz, E. C. Chapin, J. M. Martin, W. Daffer, and S. Hensley, "An assessment of the SRTM topographic products," 2005.

[11] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on.* IEEE, 2006, pp. 641–648.

[12] A. Dunkels, B. Grnvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on.* IEEE, 2004, pp. 455–462.

[13] A. Roberts, "Curvature attributes and their application to 3 D interpreted horizons," *First break*, vol. 19, no. 2, pp. 85–100, 2001.

[14] I. Mynatt, S. Bergbauer, and D. D. Pollard, "Using differential geometry to describe 3-D folds," *Journal of Structural Geology*, vol. 29, no. 7, pp. 1256–1266, 2007.

[15] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: self-configuring localization systems," in *International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK*, 2001.

[16] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A low-cost, 3d wireless sensor network testbed," in *Testbeds and Research Infrastructure. Development of Networks and Communities.* Springer, 2012, pp. 302–316.

[17] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[18] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling," in *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on.* IEEE, 2012, pp. 185–196.

[19] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *Proceedings of the IEEE National Telecommunications Conference*, vol. 4, 1978, pp. 1–4.