Adaptive Multifactor Routing with Constrained Data Sets for Autonomous Vehicle (AV) Applications

Cody Ross, Matheus Marques, Torger Miller, Mohammed Hirzallah
Haris Volos, Jonathan Sprinkle
Dept. of Electrical and Computer Engineering
University of Arizona

***Abstract*—Autonomous vehicles (AV) depend on reliable wireless communication with a remote operator or database for safe operation along a path. We plotted a map showing various signal strengths, called signal heat map. These signal heat maps represent the coverage strength in 4G and LTE network. We investigate routing the AV based on signal strength and shortest traveling distance. We consider previous routing algorithms. One novel test of the surveyed algorithms is their level of accuracy when only partial-knowledge of one or more factors is available. We constrain the knowledge of signal strength along a route and measure how close to the solution various algorithms perform. In addition, we consider the execution speed of each algorithm as the size of the map increases. By integrating heat maps of network signal strength with a distance-based routing algorithm, the best optimal path can be determined with a measurable capacity. We work with real measured signals and simulated signal maps to refine an approach defining ideal routes based on distance and signal strength.**

***Index Terms*—Autonomous vehicle routing (AV), routing with partial knowledge, Dijkstra, Bellman-Ford, Acyclic.**

I. Introduction

Autonomous vehicles (AV) depend on a reliable wireless communication with a remote operator or data network. Low signal strengths could reduce the reliability of future AV solutions. In order to ensure safety and the best optimal operation of the AV, it is desirable to route the AV based on the signal strength. Thus, we seek to investigate an adaptive routing that is based on multifactor data sets. So, we have produced few fitness formulas with different emphases. The most useful formula is the one that gives the priority to the signal strength and then to the minimum traveling distance. This results in a safe AV functioning while minimizing driving time. In this work, we use both simulated and real signal heat maps to explore various ways of routing the AV.

The most direct precursors of our research explore decentralized routing with partial knowledge resulting from network disruptions[1]. They find that routing performance is only lightly affected even when 50 percent of information sharing attempts fail. Others have also found promise in systems based on sparse traffic pattern knowledge[2].

The utility of testing algorithm performance with partial signal strengths knowledge lies in the potential for such situations to occur in practical autonomous vehicle. The use of vehicle to vehicle (V2V) or vehicle to infrastructure communication could offer real time data with varying completeness and quality. Understanding how various algorithms perform under such conditions offers good utility to be used in the AV multifactor routing. We find that with accessing 30 percent of the total signal strength map we can produce routes that are 5 percent less compared to routes produced with 100 percent signal strength knowledge.

We are mainly interested in the following distance routing algorithms: Dijkstra's Link-State Algorithm, Bellman-Ford Distance-Vector Algorithm (BF), and Directed Acyclic Algorithm via Topological Ordering (DAG). The results of our own tests confirm support for Djikstra as the fastest and most accurate algorithm for routing based on signal strength and distance.

A. Paper Organization

We have organized the paper as follows. In Section II, we explore the literature of distance-based routing algorithms and how they apply to the AV routing problem. In the Approach Section III, we detail how we have approached the problem and present the developed formula. In the Results and Discussion Section IV, we explain the performance of all three algorithms and suggest the best routing algorithm.

II. Background

Most of the literature focuses on developing routing algorithms based on traveling distance. Other literature focuses on multi-factor routing algorithms to avoid busy traffic and long traveling distance. These algorithms model the routing problem using a graph $G = (V, E)$ [3]. Street intersections (nodes) are represented by the graph vertexes V and the weights of streets as graph edges E. Denote $n = |V|$ as the number of nodes and $m = |E|$ as the number of edges. Table I presents the common distance-based routing algorithms: Dijkstra's Link-State Algorithm (DIKBA and DIKBD), Bellman-Ford's Distance-Vector Algorithm (BF), and Directed Acyclic Algorithm via Topological Ordering (DAG). As detailed in Table I, each algorithm has its own advantages and disadvantages.

Thanks to NSF and Air Force Office of Scientific Research Award #1262960 for C. Ross and T. Miller and CAPES sponsorship of M. Marques through the Brazilian Scientific Mobility Program. This paper was supported in part by the Broadband Wireless Access & Applications Center (BWAC); NSF Award #1265960

TABLE I
SUMMARY OF SHORTEST PATH ALGORITHMS

Algorithm	Advantage(s)	Disadvantage(s)	Complexity
Dijkstra Link-State Algorithm			
DIKBD[4]	• Handles larger scale graphs (arc length $>$ 1500) • Considers all weights (with loops)	• Time Consuming via Relaxation Principle • Considers only Non-Negative Weights	$O(m + n(\beta + (\frac{C}{\beta})))$
DIKBA[3]	• Handles smaller scale graphs (arc length $<$ 1500) • Considers all weights (with loops) • Terminates un-used routes during iteration process	• Time Consuming via Relaxation Principle • Considers only Non-Negative Weights	$O(m\beta + n(\beta + \frac{C}{\beta}))$
Bellman-Ford Distance-Vector Algorithm			
BF[5]	• Considers Positive and Negative Weights (with loops)	• Time Consuming via Relaxtion Principle • Does not terminate other iterations when searching for Shortest Possible Route	O(nm)
Acyclic Topological Ordering			
DAG[6]	• Operates Faster than Bellman-Ford or Dijkstra • Deletes the ignored arcs	• Less Weight Consideration • Considers only Non-Negative weights • Considers out-going weights only (No loops)	O(nm)

A. Bellman-Ford

The Bellman-Ford Algorithm[5] computes the shortest paths from a single source vertex to all other vertices in a weighted graph. In our case, weights equal the signal strength via the Relaxation Principle. The Relaxation Principle is a blind search for the shortest route possible, thereby leading to values being replaced gradually, leading to the optimal routing solution. Unfortunately, Bellman-Ford is slower than Dijkstra's Link-State Algorithm as shown in Fig.(5). This is primarily because of the Bellman-Ford algorithm's capability of processing graphs with negative weights and the amount stored as explained in [7].

B. Directed Acyclic Graph

An acyclic graph has no path that leads away from a variable only to return to that same variable. One type of acyclic graph is a Directed Acyclic Graph (DAG)[6] that considers only positive arcs (weights). No path starts and ends at the same vertex, thus there are no loops. It is possible to find the shortest and longest path from a given source vertex (source node) in DAG with linear time by processing vertices (nodes) in a topological order.

C. Dijksta Link-State Algorithm

Dijkstra's Link-State Algorithm[7] solves the single-source shortest path problem for a graph with non-negative edges via the Relaxation Principle as well. In contrary to the Bellman-Ford algorithm, the Link-State Algorithm only stores the necessary routing data in the route searching process. In other words, if a route is not the shortest possible route, then the data is terminated.

According to authors in [8], the two best algorithms for one-to-one path finding are Dijkstras Approximate Buckets (DIKBA) and Dijkstras Double Buckets (DIKBD)[8]. Also, authors found that for large paths (Network arc length over 1500) DIKBD is the best, and for smaller paths DIKBA is the best. Furthermore, authors in [4] found that DIKBD is the fastest algorithm for networks with non-negative arc lengths.

III. APPROACH

Our goal is to study the effectiveness of the developed adaptive multifactor routing scheme to route an AV with partial knowledge of the signal map. To approach this goal we test our multifactor formula with various routing algorithms by employing simulated and measured signal strengths. We route the AV based on signal strength and shortest euclidean distance. Doing so has required a way to integrate all the factors into a single fitness factor on which we could judge routes. We decided to approach fitness that indicates route inconvenience or "undesirability". This composite factor allows us to route based on a single quantity and simplifies the employment of surveyed routing algorithms. This composite factor is defined according the following formula:

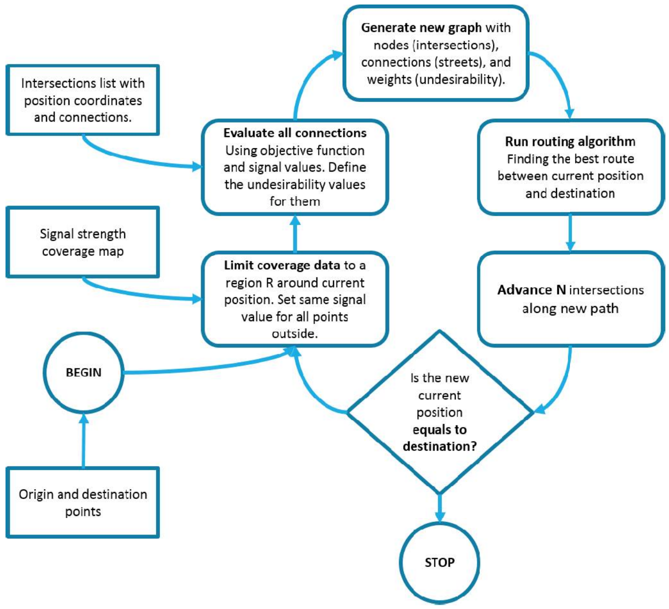

Fig. 1. Simulation flow diagram for AV routing with partial-knowledge of signal strength

$$EdgeWeight = \frac{k}{k+\beta} \cdot distance + \frac{\beta}{k+\beta} \cdot SignalStrength \quad (1)$$

where k and β, distance gain and signal average gain, are the two design thresholds used to tune the priority of signal strength as a routing factor. Using the log function allows us to prioritize the presence of a decent signal. We have used a graphical way to tune the k and β threshold values. We are still working on developing a theoretic-based method to select proper values of these thresholds and it is expected to show up in future publications.

Several fitness formulas will be needed for every particular application, given the wide variability of AV requirements and the variance in performance of various cellular technologies at different signal strengths.

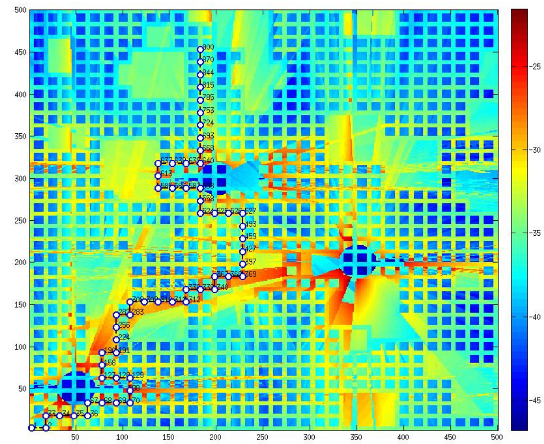

Fig. 2. Typical generated heat map with optimal route shown in dotted line

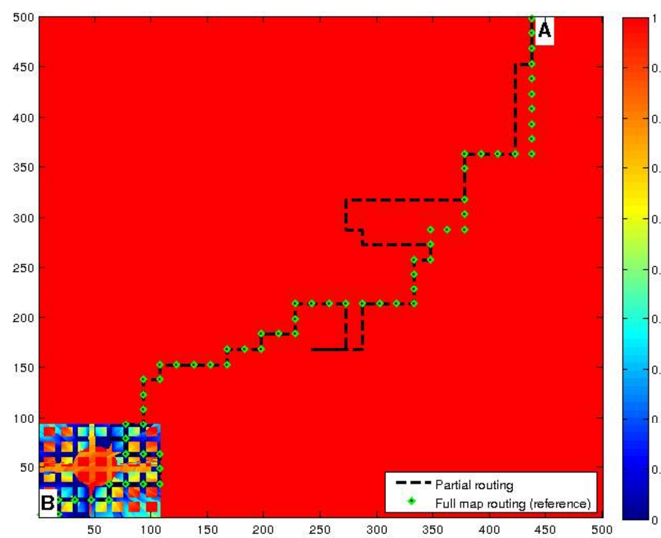

Fig. 3. Autonomous vehicle routing with partial- and full- signal knowledge

A. Simulation Methodology

For our tests of speed and path finding with partial signal knowledge, we supplement the simulated data with very large randomized maps. These randomized maps avoid some of the biases potentially presented in smaller maps corresponding to real values and offer a pure test of routing algorithm average performance. Fig.(1) illustrates the details for one simulation run.

For each simulation run we generate a map of a basic city with vertical and horizontal streets, buildings with diverse heights and some polygonal plazas. We use the empirical path loss model COST231 [9] to simulate the signal propagation. A typical generated heat map is shown in Fig (2). Table II shows the simulation parameter values. We have simulated routing algorithms performance according to three signal heat map scenarios: strong coverage, average coverage and low coverage. We have adopted the values shown in Table III for each scenario.

Fig.(3) shows a visual result for one simulation run, where the AV travels from point A to point B. We have also tested the formula with a real-recorded signal strengths, as shown in Fig. (4). We have recorded signal strengths for one of the 4G operators in Tucson, Arizona. Then, we have verified the efficiency of the proposed formula using simulation where inputs are the real measured data.

IV. Results and Discussions

We have produced two primary results. First we tested the execution speed of various shortest path algorithms to confirm the prior art that we explored in our literature review. Second, we evaluated the performance of the algorithms when only partial signal knowledge was available.

TABLE II
SIMULATION PARAMETERS

	Simulation Parameter	Value
Map specifications	Map size [km]	20 x 20
	Street widths [meter]	5
	Buildings separation distance [meter]	15
	Building heights [meter]	5, 10, 15,
Signal specifications	Channel fading	Rayleigh
	De-correlation distances (dcorr) [meter]	75, 50, 20
	Shadowing standard deviation (σ)	1, 3, 5
	Number of transmitters	100, 300,
	Transmit antenna gain	1
	Receive antenna gain	1
	Transmitted power	46 dbm
Other specifications	Number of simulation runs	1,000

TABLE III
SIGNAL COVERAGE SCENARIOS

Coverage	Strong	Average	Low
De-correlation distance [meter]	75	50	20
Shadowing standard deviation (σ)	1	3	5
Number of transmitters	500	300	100

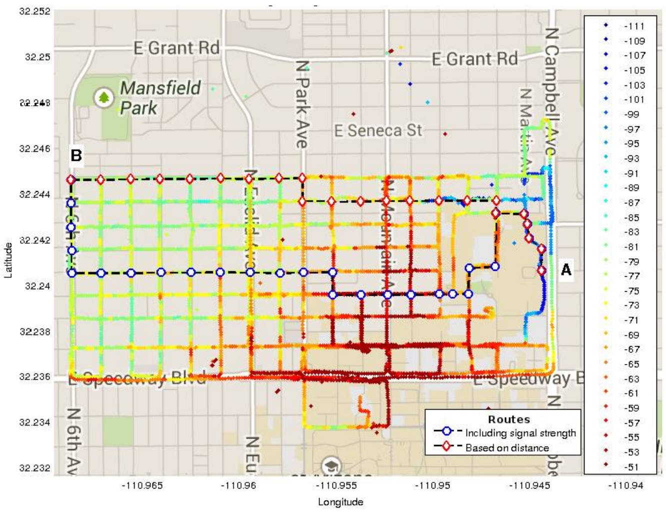

Fig. 4. Routing the AV with real measured data

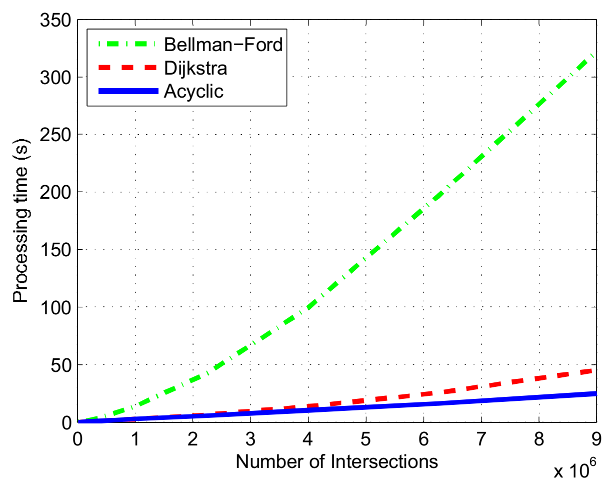

Fig. 5. Routing algorithm execution speeds

A. Execution Speed

In terms of execution speed, we may see in Fig.(5) that Djikstra is the clear winner over Bellman-Ford as the number of intersections increases, with Bellman-Ford taking an order of magnitude longer for very large numbers of intersections. In addition, while Acyclic outperforms Djikstra and Bellman-Ford in terms of speed, it produces less optimal solutions. Acyclic does give some consideration to weights, but first minimizes the number of nodes traversed by a route, and then minimizes the weights to the extent possible without increasing the number of nodes. Consequently, it will rarely match the performance of Djikstra and Bellman-Ford in routing based on weighted values. These results mirror the earlier information presented from the literature and confirm the suitability of Djikstra as a speedy, highly optimal algorithm [8]. We include Acyclic throughout our results as a useful comparison to the identical solutions produced by Bellman-Ford and Djikstra.

B. Partial Knowledge Effects

We explore how the algorithms perform with partial data availability. Fig.(6) presents algorithms' performance for strong signal coverage.

1) Dijkstra and Bellman-Ford Performance: As shown in Fig.(6) demonstrates demonstrates that for Djikstra and Bellman-Ford there are rapid gains initially with increasing signal knowledge followed by a plateau at a partial signal knowledge size of 10 percent. A curve such as this is typical and expected in applications such as these where increased data tends to offer diminishing returns. This data confirms our expectations that routing autonomous vehicles based on factors of which there is only partial knowledge may be highly effective. Further research would be useful to confirm this trend for other routing data, such as the expected fuel consumption on routes with varying elevation and speed limit data. The ability to use only partially known factors offers great promise in reducing the amount of data transmission and storage required to implement effective autonomous vehicle routing systems. The utility of partially known factors is particularly exiting for vehicle to vehicle communications in which each vehicle only has limited information for a small surrounding area. The utility of partial factors also offers the possibility of reducing load on vehicle to infrastructure communications networks as routing may be 95 percent effective with only a tenth of the data required for 100 percent optimal routing.

2) Acyclic Performance: One exception to our results is the performance of Acyclic where we notice a flat response to increased signal knowledge, as shown in Fig.(6). The reason for the flat response lies in how Acyclic considers signal data. Acyclic first finds the route with the smallest number of nodes. Only then does it optimize for our weights that combine signal strength and distance. What this process means is that Acyclic will only alter its path based off signal strength

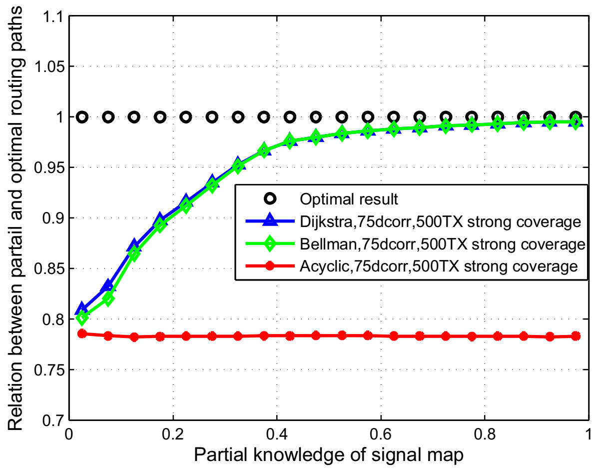

Fig. 6. Routing algorithms performance with partial signal knowledge

if there are multiple paths with the least possible number of nodes. Consequently, as signal strength is always a secondary consideration for Acyclic, we would expect it to vary its path recommendations relatively little in response to increased signal data compared to Djikstra and Bellman-Ford which fully consider signal weights. The data confirms these expectations. We plainly see little response by Acyclic to increased signal data. Thus, what appears a potential contradiction to our results is merely a confirmation of our expectations for algorithm performance.

3) Signal Coverage Effects: Fig.(7) - (9) demonstrate the performance per each routing algorithms for various coverage scenarios. As shown, routing algorithms are slightly affected by the signal coverage when performing with partial knowledge. The best performance is noticed for strong signal coverage.

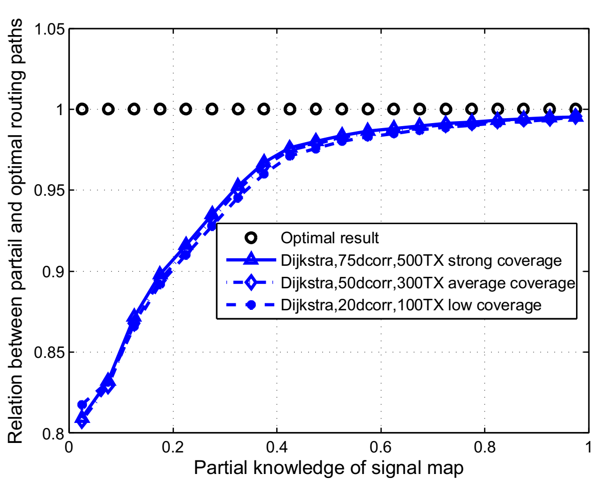

Fig. 7. Dijkstra routing algorithm performance with partial signal knowledge

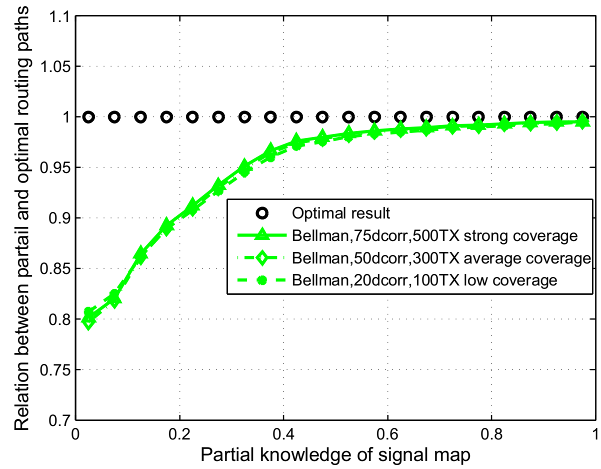

Fig. 8. Bellman routing algorithm performance with partial signal knowledge

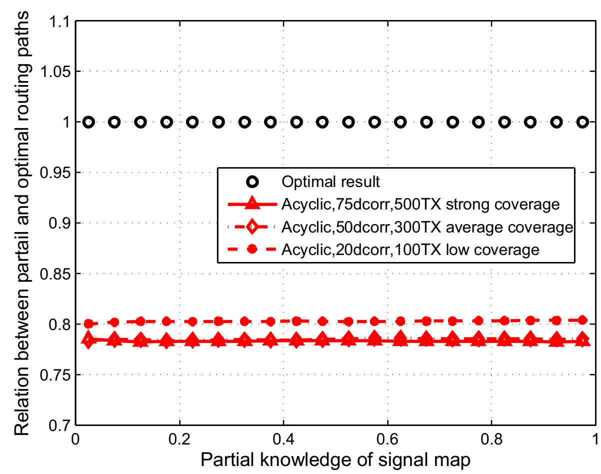

Fig. 9. Acyclic routing algorithm performance with partial signal knowledge

V. Conclusion

Autonomous vehicles routing based on signal strength enhances safety and reliability. As we have seen, partial knowledge of signal strength can offer substantial improvements to autonomous vehicle performance in the form of improved routing. Successful autonomous routing with partial signal strength knowledge overcomes the challenges associated with data transfer between the vehicle and infrastructure, and also reduces the data that needs to be collected. We have examined previously developed routing algorithms with multifactor routing. We have found that with 30 percent of signal strength knowledge the autonomous vehicle will follow a route that is 5 percent close to the optimal one.

References

[1] B. Scholz-Reiter and C. Zabel, "Influences of communication disruptions on decentralized routing in transport logistics," in *Recent Researches in Mechanics. Proceedings of the 4th WSEAS International Conference on Urban Planning and Transportation (UPT'11)*, p. 265.

[2] M. de B do Amarante and A. L. Bazzan, "Agent-based simulation of mobility in real-world transportation networks: effects of acquiring information and replanning en-route," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1351–1352.
[3] F. B. Zhan, "Three fastest shortest path algorithms on real road networks: Data structures and procedures," *Journal of geographic information and decision analysis*, vol. 1, no. 1, pp. 69–82, 1997.
[4] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Mathematical programming*, vol. 73, no. 2, pp. 129–174, 1996.
[5] L. R. Ford, "Network flow theory." 1956.
[6] D. J. Pearce and P. H. Kelly, "A dynamic algorithm for topologically sorting directed acyclic graphs," in *Experimental and Efficient Algorithms*. Springer, 2004, pp. 383–398.
[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[8] F. B. Zhan and C. E. Noon, "Shortest path algorithms: an evaluation using real road networks," *Transportation Science*, vol. 32, no. 1, pp. 65–73, 1998.
[9] G. L. Stüber, "Path loss models," in *Principles of mobile communication*. Springer, 2011, pp. 132–146.