engineering proceedings

Proceeding Paper

A Web Platform for Generating Optimization-Ready V2V/V2I Connectivity Data from SUMO Vehicular Scenarios †

Edison Xavier Espinel Sevilla [1], Soraya Lucía Sinche Maita [1], Martha Cecilia Paredes-Paredes [1] and Luis Urquiza-Aguiar [1,2,*]

[1] Departamento de Electrónica, Telecomunicaciones y Redes de Información, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito 170525, Ecuador; edison.espinel@epn.edu.ec (E.X.E.S.); soraya.sinche@epn.edu.ec (S.L.S.M.); cecilia.paredes@epn.edu.ec (M.C.P.-P.)
[2] MODEMAT Foundation for Mathematical Modeling and Education, Quito 170109, Ecuador
* Correspondence: luis.urquiza@epn.edu.ec
† Presented at the XXXIV Conference on Electrical and Electronic Engineering (JIEE 2026), Quito, Ecuador, 26–30 October 2026.

Abstract

Roadside-unit (RSU) deployment models need data that tell which vehicles can reach each candidate RSU, in which mobility snapshot, and with how many hops. Producing these data is not trivial: it means combining traffic mobility, urban geometry, direct-link evaluation, graph processing, and optimization-oriented data structures. This paper presents a web platform that turns a geographical area and a SUMO mobility scenario into an optimization-ready connectivity dataset. The backend integrates OpenStreetMap and SUMO tools, filters candidate RSU positions, evaluates binary V2V and V2I links from communication range and building obstruction, and computes cumulative and minimum-hop connectivity through a matrix recurrence. A formal proposition justifies the recurrence used by the pipeline: it recovers every vehicle–RSU relation within a given hop limit, while first-arrival matrices identify the minimum hop count. A Streamlit frontend supports configuration, execution, visualization, and export. On the Historic Centre of Quito, varying the minimum junction degree and the clustering radius keeps between 14 and 393 of 1211 eligible junctions as candidates (about 68% to 99% reduction). The workflow separates mobility simulation from infrastructure optimization, so the same dataset can feed different exact or heuristic deployment models.

Keywords: VANET; V2V connectivity; V2I connectivity; SUMO; multi-hop connectivity; roadside units; preprocessing; facility location; operations research

1. Introduction

Vehicular ad hoc networks (VANETs) enable direct vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication with fixed roadside units (RSUs), usually installed at intersections. Where these units are placed affects coverage, path length, infrastructure cost, and network performance, so RSU deployment is often stated as a facility-location problem and solved with exact, heuristic, or metaheuristic methods. Urban V2X simulation studies have also examined how communication range and RSU placement affect road-network coverage, as illustrated by the Xanthi case study [1].

These models need connectivity information before the optimization starts: which vehicles can reach each candidate RSU in each snapshot, and the minimum number of

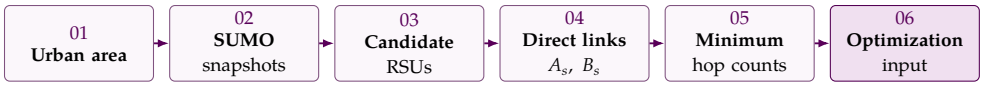
Check for updates

Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

relay hops. Producing it is not a minor step, as it combines traffic simulation, vehicle-position extraction, urban geometry, range checks, obstacle detection, graph processing, and conversion into the structures a solver accepts. Yet many deployment studies treat these data as already available. The software side is similar: SUMO and Python libraries are usually glued together with ad hoc scripts, which hurts reproducibility and reuse across formulations.

This work fills that gap with a web-based preprocessing layer that follows the sequence



Its main output is not just a set of connectivity matrices, but a traceable dataset that keeps the relation between geographical positions, SUMO identifiers, matrix indices, snapshots, hop counts, and optimization identifiers. The contributions are: (i) a web platform that integrates a Streamlit frontend with a Python/SUMO backend in a reproducible workflow; (ii) automatic generation of direct V2V and V2I matrices under a binary communication-range and line-of-sight (LoS) model; (iii) a configurable two-stage method that reduces the candidate RSU set using junction degree and spatial clustering; (iv) computation of cumulative and minimum-hop vehicle–RSU matrices; (v) export of optimization-ready dense structures and sparse tuples, including an artificial no-service option per vehicle; (vi) a formal justification of the recurrence and minimum-hop representation used in the data-generation pipeline; and (vii) a controlled analysis of vehicular workload and of how candidate filtering changes potential reachability, dataset size, and preprocessing time. The optimization model that picks the final deployment is out of scope: the goal is to generate its input automatically and independently of any formulation.

2. Related Work

RSU deployment has been studied through coverage, cost, connectivity, and quality-of-service objectives, using exact optimization, heuristics, and metaheuristics [2]. Relevant approaches include direct and multi-hop access [3], scenario-dependent connectivity [4], urban planning based on vehicle densities [5], and information-timeliness requirements [6]. Recent studies also examine RSU spacing under non-uniform vehicle distributions [7] and joint deployment and adaptive beam steering using six-dimensional movable antennas [8]. These formulations differ in their objectives and communication assumptions but require candidate locations and vehicle–RSU service relationships.

SUMO provides microscopic mobility simulation over road networks obtained, for example, from OpenStreetMap [9,10]. TrafficModeler extensions combine OSM, SUMO, OMNeT++, and Veins to facilitate VANET scenario construction and evaluation [11]. STGT provides graphical and command-line interfaces for traffic-scenario generation, simulation, and mobility-statistics extraction [12]. These documented workflows facilitate simulation preparation and evaluation. The additional data-preparation task addressed here is to convert mobility snapshots into indexed vehicle–candidate-RSU reachability relations for deployment models.

For coverage- and hop-based formulations, these relations must identify which candidate RSUs each active vehicle can reach in each snapshot and the corresponding minimum hop count. The proposed platform automates this transformation by combining candidate filtering, range- and obstacle-based direct-link construction, cumulative and first-arrival reachability matrices, and dense-matrix and sparse-tuple exports. It also preserves the correspondence between geographical locations, SUMO identifiers, matrix indices, and optimization identifiers. The specific contribution relative to the cited simulation workflows

https://doi.org/10.3390/engproc2026158021

is this integrated production of traceable deployment inputs. No new placement algorithm or graph-search method is claimed; the contribution lies in connecting these established components within a reproducible preprocessing workflow.

3. Web Platform and Connectivity-Data Generation

3.1. Platform Architecture and Workflow

The platform follows a simplified model–view–controller organization. The Streamlit frontend captures the area and the parameters, launches the workflow, and displays maps, matrices, and summary indicators, while the Python backend prepares the SUMO scenario, extracts snapshots, filters candidate RSUs, evaluates direct links, computes multi-hop reachability, and exports the dataset; an orchestrator coordinates the modules and stores each run's configuration. As shown in Figure 1, OSM data become a SUMO network and building polygons through `netconvert` and `polyconvert`, `randomTrips.py` generates the demand, and the simulation produces the FCD snapshots that feed the connectivity engine.

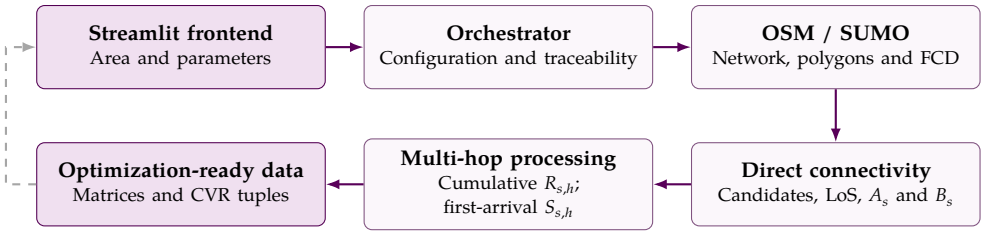


Figure 1. Platform architecture and data-generation workflow. Solid arrows show processing order; the dashed return shows results sent back for visualization, not an additional optimization stage.

The configurable parameters are grouped into mobility settings (bounding box, simulation duration, vehicle generation, snapshot interval, and random seed), candidate-selection settings (minimum junction degree $g_{\min}$ and clustering radius ρ), communication settings (V2V/V2I range and building obstruction), and the maximum hop count H. In the case study, the interface used a V2V/V2I range of 300 m, a 120 s snapshot interval, $g_{\min} = 4$, and $\rho = 20$ m as defaults.

3.2. Candidate RSU Selection

SUMO can produce a large set of junctions, including intermediate road points, internal nodes, and closely located intersections; using all of them as candidates enlarges the search space and adds nearly equivalent positions. The platform therefore applies a two-stage reduction. First, directed and parallel SUMO edges are reduced to unique unordered pairs of non-internal junctions; the degree of a junction is therefore the number of distinct neighboring junctions in this undirected simple graph. Nodes with degree below $g_{\min}$ are discarded. Second, the survivors are ordered by decreasing degree and, when degrees are equal, by increasing SUMO junction identifier. Greedy spatial clustering then accepts a node only if its distance to every previously accepted candidate exceeds ρ. Coordinates are projected to UTM zone 17S before clustering, so ρ and all pairwise distances are measured in meters. Because higher-degree nodes are processed first, they represent their local groups, while the identifier tie-break makes repeated runs deterministic. The complete candidate-filtering procedure is summarized in Algorithm 1.

This is a preprocessing heuristic, not an optimal RSU-placement algorithm. Its purpose is to build a manageable candidate set before the deployment model is solved. Since it may drop a location that would later be useful, the sensitivity of the retained set to $(g_{\min}, \rho)$ is reported in Section 4 instead of hiding these choices as fixed constants.

https://doi.org/10.3390/engproc2026158021

Algorithm 1: Candidate RSU filtering by degree and greedy spatial clustering

Input: Junction set J, unique unordered junction-pair set E_u, minimum degree $g_{\min}$, clustering radius ρ in projected metres
Output: Candidate RSU set C

Initialize $\deg(j) \leftarrow 0$ for all $j \in J$;
foreach $\{u, v\} \in E_u$ **do**
  increment $\deg(u)$ and $\deg(v)$;
$F \leftarrow \{j \in J : \deg(j) \geq g_{\min}\}$;
Sort F by non-increasing degree and then by increasing SUMO identifier;
$C \leftarrow \emptyset$;
foreach $j \in F$ **do**
  if $C = \emptyset$ ***or*** $\min_{c \in C} \mathrm{dist}(j, c) > \rho$ **then**
    $C \leftarrow C \cup \{j\}$;
return C;

3.3. Direct and Multi-Hop Connectivity

For each snapshot $s \in \mathcal{S}$, let $\mathcal{V}_s = \{v_1, \ldots, v_{n_s}\}$ be the active vehicles and $\mathcal{R} = \{r_1, \ldots, r_m\}$ the candidate RSUs. The binary V2V matrix $A_s \in \{0,1\}^{n_s \times n_s}$ has $(A_s)_{ij} = 1$ when v_i and v_j are within the V2V range and no building polygon blocks the segment joining them; under the bidirectional-link model it is symmetric. The binary V2I matrix $B_s \in \{0,1\}^{n_s \times m}$ has $(B_s)_{ik} = 1$ when v_i can directly reach r_k under the V2I range and LoS. Obstruction uses a 2-D segment–polygon intersection test, so the result is a geometric binary approximation; it does not model attenuation, fading, interference, or packet delivery. The binary entries express link admissibility under the selected communication criterion. Users may configure the ranges using external link-budget calculations, propagation models, or measurements appropriate to the technology, operating conditions, and intended per-link reliability level. The platform applies these ranges but does not itself estimate the resulting packet-delivery probability.

A vehicle without direct V2I access may still reach an RSU through other vehicles. Define the augmented adjacency matrix $\widetilde{A}_s = \beta(A_s + I)$, where $\beta(X)$ maps every positive entry of X to one and every zero entry to zero. Cumulative connectivity is

$$R_{s,1} = B_s, \qquad R_{s,h} = \beta\big(\widetilde{A}_s R_{s,h-1}\big), \quad h = 2, \ldots, H. \tag{1}$$

The identity term keeps the paths already found at smaller hop counts, which gives $R_{s,1} \leq R_{s,2} \leq \cdots \leq R_{s,H}$ elementwise. The first-arrival matrices are $S_{s,1} = R_{s,1}$ and $S_{s,h} = R_{s,h} - R_{s,h-1}$ for $h \geq 2$.

Proposition 1. *For any snapshot s, $(R_{s,h})_{ik} = 1$ if and only if there is a path from vehicle v_i to candidate RSU r_k using at most h communication hops, where the final hop is V2I and every previous hop is V2V.*

Proof. By induction on h. For $h = 1$, $R_{s,1} = B_s$, so an entry is one exactly for a direct V2I link. Assume the statement holds for $h - 1$. The entry $(\widetilde{A}_s R_{s,h-1})_{ik}$ is positive if there is an index j with $(\widetilde{A}_s)_{ij} = 1$ and $(R_{s,h-1})_{jk} = 1$. If $j = i$, the identity term keeps a path of at most $h - 1$ hops. If $j \neq i$, then $(A_s)_{ij} = 1$ adds one V2V hop from v_i to v_j before a path of at most $h - 1$ hops from v_j to r_k. Conversely, every path of at most h hops either already has at most $h - 1$ hops or begins with one V2V hop followed by such a path. Binarization therefore gives exactly the stated reachability relation. □

https://doi.org/10.3390/engproc2026158021

Corollary 1. *For $h \geq 2$, $(S_{s,h})_{ik} = 1$ if and only if the minimum number of hops from v_i to r_k is exactly h; for $h = 1$, $S_{s,1} = B_s$ identifies the direct connections.*

The corollary follows from Proposition 1 and the monotonicity of $R_{s,h}$: each reachable vehicle–RSU pair appears for the first time at a single hop value only. Figure 2 shows a compact example. Under ordinary dense matrix multiplication, computing the $H - 1$ propagation steps costs $O((H-1)n_s^2 m)$ per snapshot. Storing A_s and all cumulative and first-arrival matrices requires $O(n_s^2 + Hn_s m)$ space. The recurrence is used because these matrices map directly to the minimum-hop tuple export while preserving vehicle and RSU indexing. A hop-bounded BFS using the same snapshot graph and relay rules would recover the same minimum-hop relations; this mathematical equivalence does not imply equivalent runtime or memory consumption.

R_3	r_1	r_2	r_3
v_1	1	1	1
v_2	1	1	1
v_3	1	1	1
v_4	0	0	0

S_2	r_1	r_2	r_3
v_1	0	0	1
v_2	1	1	0
v_3	0	0	1
v_4	0	0	0

S_3	r_1	r_2	r_3
v_1	0	1	0
v_2	0	0	0
v_3	1	0	0
v_4	0	0	0

Figure 2. Illustrative cumulative (R_3) and first-arrival (S_2, S_3) matrices for $H = 3$. Purple cells in R_3 indicate reachability within at most three hops, whereas purple cells in S_2 and S_3 identify first arrivals at exactly two and three hops, respectively. Vehicle v_4 reaches no real RSU within the hop limit.

3.4. Optimization-Oriented Export and Cardinality

Let $K_s = \sum_{h=1}^{H} \mathrm{nnz}(S_{s,h}) = \mathrm{nnz}(R_{s,H})$ be the number of reachable real vehicle–RSU pairs in snapshot s. The export module creates one tuple for each real pair at its minimum hop and one artificial no-service tuple for every active vehicle:

$$\mathrm{CVR}_s = \{\langle s, h, i, k\rangle : (S_{s,h})_{ik} = 1\} \cup \{\langle s, H+1, i, r_\infty\rangle : i \in \mathcal{V}_s\}.$$

The artificial RSU r_∞ (identifier 0) is not only physical disconnection: it is the optimization option of leaving a vehicle unserved when real candidates are not deployed, are saturated, or are rejected by cost or capacity constraints. It is therefore included even when the vehicle can reach real RSUs and is penalized at hop $H + 1$. The cardinality is $|\mathrm{CVR}_s| = K_s + n_s$, with $n_s \leq |\mathrm{CVR}_s| \leq n_s(m+1)$ for $m \geq 1$; the hop limit H does not multiply the real tuples since each reachable pair is exported once, at its minimum hop. The dataset can be written to an OPL/`.dat` file or kept in memory for other solvers.

Across all snapshots, define

$$N = \sum_{s \in \mathcal{S}} n_s, \qquad K = \sum_{s \in \mathcal{S}} \mathrm{nnz}(R_{s,H}), \qquad |\mathrm{CVR}| = K + N.$$

Here N counts vehicle–snapshot instances rather than distinct vehicle identifiers. Consequently, a valid export must contain exactly K real tuples and exactly N artificial tuples. This identity provides a direct integrity check for the generated dataset.

4. Experimental Results

4.1. Experimental Settings

The evaluation uses the Historic Centre of Quito, Ecuador, with the same archived OSM/SUMO network and building polygons as the original experiment. The extract spans about $-78.535°$ to $-78.497°$ longitude and $-0.266°$ to $-0.211°$ latitude (UTM zone 17S, WGS84), and was downloaded on 5 June 2026. It contains 1211 eligible junctions. The original candidate counts were recovered: U 1211, A 393, B 366, C 278, and D 14. Algorithm 1 was applied with unique undirected neighbours and an identifier tie-break; the public repository preserves both the source snapshot and the paper-specific filtering adapter.

The archived trace also reproduced the previous $N = 201$ and $|\mathrm{CVR}| = 1628$. However, inspection of its route file found 150 routed vehicles, whereas 132 was the number of distinct identifiers observed in the sampled FCD, not the number of generated trips. To make demand levels explicit and comparable, all three traces were regenerated with targets of 132, 660, and 1320 trips. The reference results below therefore refer to the new low-demand trace, not the historical one.

To characterize the implemented pipeline under increasing vehicular workloads, three demand levels were evaluated over the same Quito road network while keeping the candidate set, communication ranges, hop limit, snapshot interval, and software environment fixed. The candidate-filtering analysis used the same low-demand mobility trace and compared the unfiltered eligible set U with configurations A–D. Seven unique executions were required because configuration C at low demand was shared by both analyses. The generation period was 3660/target trips seconds, with seed 42 passed explicitly to both `randomTrips.py` and SUMO. All runs used $0 \le t < 3660$ s and the same 31 sampling instants $t = 0, 120, \ldots, 3600$ s, including any empty samples. Both direct ranges were 300 m and $H = 3$. Only demand changed in the density study; U and A–D used identical low-demand FCD and identical V2V matrices. The unfiltered set U excludes internal and dead-end junctions but applies neither degree nor clustering thresholds.

Experiments ran with SUMO 1.26.0, Python 3.10.11, and NumPy 2.2.6 on an AMD Ryzen 7 7735H computer with nominally 16 GB RAM under Windows 11. SUMO used one simulation thread. Elapsed times were measured with `time.perf_counter`: T_{links} includes geometric V2V/V2I evaluation and aligned dense A_s, B_s construction; T_{hops} includes identity augmentation and the cumulative and first-arrival recurrences through $H = 3$. V2V was recomputed with the same procedure in each filtering run and checked for equality, so no run benefited from reusing a previously timed matrix. Download, SUMO simulation, XML parsing, candidate filtering, file export, and map rendering are outside these two timers. Thus, T_{pre} below denotes the two timed connectivity stages, not the entire end-to-end workflow.

For each run, potential reachability and the mean number of reachable candidates are

$$Q_{\mathrm{reach}} = \frac{100}{N} \sum_{s} \sum_{i=1}^{n_s} 1\left\{ \sum_{k=1}^{m} (R_{s,H})_{ik} > 0 \right\}, \qquad \bar{r} = K/N.$$

The first metric counts instances that reach at least one real candidate; the second counts available RSU alternatives, including zero for unreachable instances. Neither metric is the coverage of a selected deployment. The public repository includes one summary CSV, commands, seed, source and input hashes, and seven exported datasets. Every candidate is retained in the export, including those with no contacts. No deployment solver was run. These are single-seed controlled workload tests, not independent statistical replications; no variability or memory measurements are claimed.

4.2. Candidate-Filtering Results

Table 1 extends the candidate-count analysis with reachability, alternative RSUs, tuple counts, and connectivity-processing time.

Configuration C retains 278 of 1211 eligible candidates, reducing the exported tuple count from 6090 to 1598 and the measured connectivity-processing time from 15.065 to 4.131 s. Potential reachability decreases from 100.00% to 98.88%, while the mean number of reachable candidates falls from 33.02 to 7.93, indicating a loss of connectivity alternatives. Configuration D reaches only 45.25% of instances, illustrating the connectivity cost of aggressive filtering in this scenario. These measurements quantify preprocessing trade-offs; they do not establish preservation of an optimal deployment solution.

https://doi.org/10.3390/engproc2026158021

Table 1. Candidate filtering on the same low-demand mobility trace ($X = 1211$, $N = 179$, $H = 3$). Reachability is potential access to candidates, not coverage after deployment. $T_{\text{pre}} = T_{\text{links}} + T_{\text{hops}}$.

Cfg.	$g_{\min}/\rho$ (m)	Y	Red. (%)	Q_{reach} (%)	K/N	|CVR|	T_{pre} (s)
U	—	1211	0.0	100.00	33.02	6090	15.065
A	3/15	393	67.5	99.44	10.20	2005	5.625
B	3/20	366	69.8	99.44	8.98	1786	5.237
C	4/20	278	77.0	98.88	7.93	1598	4.131
D	5/25	14	98.8	45.25	0.68	300	0.544

A–B changes only the clustering radius at $g_{\min} = 3$; B–C changes only the degree threshold at $\rho = 20$ m. D changes both and cannot isolate either effect. The comparison is scenario-specific: fewer candidates do not by themselves imply a better deployment, and these candidate sets are not assumed to be nested. Figure 3 shows the unchanged candidate set C over the archived map.

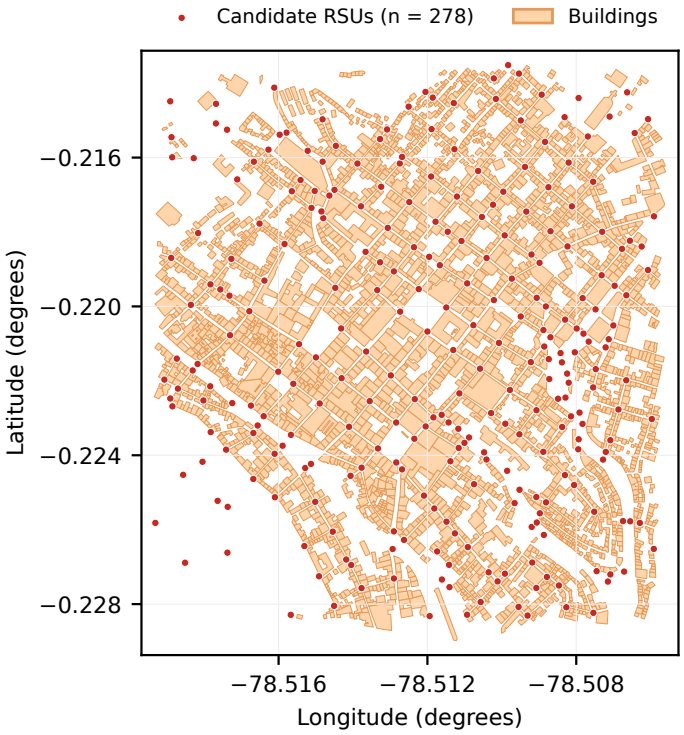


Figure 3. Candidate RSUs for configuration C ($g_{\min} = 4$, $\rho = 20$ m; $n = 278$, red) over building footprints (orange) in the Historic Centre of Quito. Coordinates are recovered using the projection and offset stored in the archived SUMO network.

4.3. Vehicular-Workload Results

With configuration C fixed, a 10.4-fold increase in mean active vehicles (5.77 to 59.87) increased direct-link construction time 9.1-fold (4.128 to 37.701 s) and the number of exported tuples 18.4-fold (1598 to 29,383); multi-hop processing took at most 0.044 s. Direct-link construction dominated the two measured connectivity stages in all three runs. These single-seed measurements characterize the tested workloads only; evaluation across different network sizes, road topologies, and seeds remains future work, and no general scaling law is claimed. Table 2 reports the complete measurements for the three demand levels.

Table 2. Controlled demand increase with configuration C (278 candidates, 31 snapshots, seed 42). Trips are vehicles present in the generated route file; active vehicles are counted in the FCD snapshots. Each time is one measured run.

Demand	Trips	Active Mean/Max.	N	T_{links} (s)	T_{hops} (s)	|CVR|
Low	132	5.77 / 7	179	4.128	0.003	1598
Medium	660	29.55 / 37	916	17.270	0.022	10181
High	1320	59.87 / 71	1856	37.701	0.044	29383

4.4. Reference Connectivity and Export

Table 3 updates the reference metrics to the new C low-demand trace. Counts and timings from the archived experiment are not mixed with these results.

Table 3. Connectivity and export metrics for the new reference configuration C (278 candidates, seed 42, $H = 3$).

Metric	Calculation and Meaning	Result
Hop limit	Includes the final V2I hop	$H = 3$
Direct real pairs	$M_1 = \sum_s \text{nnz}(S_{s,1})$	1242
New real pairs by hop	$M_2 = \sum_s \text{nnz}(S_{s,2})$; $M_3 = \sum_s \text{nnz}(S_{s,3})$	154; 23
Reachable real pairs	$K = \sum_h M_h = \sum_s \text{nnz}(R_{s,H})$	1419
Vehicle–snapshot instances	$N = \sum_s n_s$; distinct sampled vehicle IDs: 113	179
Exported tuple set	$\|\text{CVR}\| = K + N$	1598
Output size	Plain-text dataset, 1024 bytes per KiB	60.8 KiB

Multi-hop forwarding adds $M_2 + M_3 = 177$ real pairs to the 1242 direct ones. These are additional vehicle–RSU alternatives, not necessarily additional connected vehicles. The identity $|\text{CVR}| - K = N$ holds exactly ($1598 - 1419 = 179$) and was checked in all seven runs, together with tuple uniqueness and the number of real first-arrival pairs at each snapshot and hop. Table 4 gives tuples read from the new reference export.

Table 4. Verified tuples from the new C low-demand export and their inverse mappings.

$\langle s, h, i, k \rangle$	Meaning	t (s)	Vehicle	SUMO Junction
$\langle 1, 1, 0, 22 \rangle$	1 hop	0	V0	11135269234
$\langle 4, 2, 6, 48 \rangle$	2 hops	360	V6	267037293
$\langle 8, 3, 25, 91 \rangle$	3 hops	840	V25	268819798
$\langle 1, 4, 0, 0 \rangle$	artificial	0	V0	—

This first implementation supplies potential connectivity for coverage- and hop-based RSU deployment models using configurable ranges and 2-D LoS screening. Alternative link-admission criteria derived from propagation, fading, or packet-reception models could populate A_s and B_s without changing the snapshot minimum-hop recurrence or tuple-export logic. The current criterion may exclude viable NLoS links, while the reliability of retained LoS links depends on the selected range and operating conditions; the reported connectivity is therefore conditional on these assumptions.

Each snapshot describes simultaneous paths, not store-carry-forward routes or relay latency. The measurements concern one network, one seed, 31 snapshots, and $H = 3$; other topologies, hop limits, memory behaviour, and comparisons with BFS or sparse kernels remain untested. Assessing deployment cost, capacity-constrained service, and solver time

requires a specified deployment objective and constraints. Such a comparison between filtered and unfiltered candidate sets was not performed and remains future work.

5. Conclusions and Future Work

This paper presented a web platform that turns a geographical area and SUMO mobility output into optimization-ready V2V/V2I connectivity data, combining candidate RSU generation, direct-link evaluation with building obstruction, cumulative and minimum-hop matrix processing, traceable identifiers, and export.

A formal result justifies the recurrence used by the pipeline: it recovers each vehicle–RSU relation reachable within the hop limit, and the first-arrival matrices assign each real pair its minimum hop; the exported set adds one artificial no-service option per vehicle, an optimization decision rather than mere disconnection. On the Historic Centre of Quito, filtering keeps 14–393 of 1211 eligible junctions depending on $(g_{\min}, \rho)$, a preprocessing trade-off that reduces the candidate set without guaranteeing a better deployment. The new controlled tests quantify this trade-off: configuration C reaches 98.88% of low-demand instances, compared with 100.00% for all eligible candidates. Three demand levels show how measured processing time and exported data size grow on this fixed network, without establishing general scalability.

Future work includes benchmarking the recurrence against breadth-first search, sparse-matrix and Boolean-semiring implementations in terms of runtime and memory across different vehicle counts and hop limits, dynamic tuple generation inside the optimization method (e.g. decomposition or column generation), technology-specific link-admission criteria based on propagation and packet-reception models, including Markovian representations of packet-error dynamics [13], extensions with path-level attributes and feasibility checks for end-to-end latency and reliability, and measuring the effect of filtering on deployment cost, coverage, capacity, and solution time.

Author Contributions: Conceptualization, L.U.-A.; methodology, E.X.E.S., M.C.P.-P. and L.U.-A.; software, E.X.E.S.; validation, E.X.E.S. and M.C.P.-P.; investigation, E.X.E.S.; data curation, E.X.E.S.; visualization, E.X.E.S.; writing—original draft preparation, E.X.E.S.; writing—review and editing, S.L.S.M., M.C.P.-P. and L.U.-A.; supervision, L.U.-A.; project administration, S.L.S.M. and L.U.-A.; funding acquisition, S.L.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Escuela Politécnica Nacional through the research project "SmartCityNet: IoT y Redes Vehiculares para la Optimización y Transformación Urbana" (grant no. PIGR-24-02).

Data Availability Statement: The source code, generated mobility and connectivity data, experimental results, dependency information, and documentation supporting this study are publicly available at https://github.com/xtremeecologytwo/smartcitynet-vanet, accessed on accessed on 1 October 2026.

Acknowledgments: Edison Xavier Espinel Sevilla conducted this work as a student research intern within the activities of the research project "SmartCityNet: IoT y Redes Vehiculares para la Optimización y Transformación Urbana" (PIGR-24-02) at Escuela Politécnica Nacional. During manuscript preparation, the authors used Writefull (Writefull Ltd., London, UK, 2026) for language editing and ChatGPT (GPT-5, OpenAI, San Francisco, CA, USA, 2026) for structural, terminological, and formatting support. The authors reviewed and validated all outputs and take full responsibility for the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bazinas, E.; Gregoriades, A.; Raspopoulos, M.; Georgiades, M. V2X communication coverage analysis for connected vehicles in intelligent transportation networks: A case study for the city of Xanthi, Greece. *arXiv* **2023**, arXiv:2401.00465. [CrossRef]
2. Guerna, A.; Bitam, S.; Calafate, C.T. Roadside Unit Deployment in Internet of Vehicles Systems: A Survey. *Sensors* **2022**, *22*, 3190. [CrossRef]
3. Wu, T.-J.; Liao, W.; Chang, C.-J. A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks. *IEEE Trans. Commun.* **2012**, *60*, 2295–2303. [CrossRef]
4. Urquiza-Aguiar, L.; Tripp-Barba, C.; Aguilar-Igartua, M. A Stochastic Optimization Model for the Placement of Road Side Units. In *Proceedings of ADHOC-NOW*; Springer: Cham, Switzerland, 2016.
5. Astudillo León, J.P.; Busson, A.; de la Cruz Llopis, L.J.; Begin, T.; Boukerche, A. Strategic Deployment of RSUs in Urban Settings: Optimizing IEEE 802.11p Infrastructure. *Ad Hoc Netw.* **2024**, *163*, 103585. [CrossRef]
6. Liang, B.; Xu, X.; Lu, W.; Ran, B. Optimizing RSU Deployment in VANETs: A Branch-and-Benders Decomposition Approach Considering Information Timeliness Requirements. *IEEE Trans. Intell. Transp. Syst.* **2026**, *27*, 4255–4270. [CrossRef]
7. Zhou, J.; Yu, X.; Geng, J.; Zhang, Y.; Zhang, L.; Mou, Z. Highway roadside unit deployment under uneven distribution evolution of intelligent connected vehicles. *J. Highw. Transp. Res. Dev. (Eng. Ed.)* **2026**, *20*, 67–77. [CrossRef]
8. Wu, X.; Huang, R.; Wei, C.; Chu, X.; Chen, J.; Zhao, S. Towards Dynamic V2X Infrastructure: Joint Deployment and Optimization of 6DMA-Enabled RSUs. *Sensors* **2026**, *26*, 388. [CrossRef]
9. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic Traffic Simulation using SUMO. In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
10. OpenStreetMap Contributors. Available online: https://www.openstreetmap.org (accessed on 20 July 2026).
11. Arellano, W.; Mahgoub, I. TrafficModeler Extensions: A Case for Rapid VANET Simulation Using OMNeT++, SUMO, and Veins. In Proceedings of the 10th International Conference on High Capacity Optical Networks and Emerging/Enabling Technologies, Magosa, Cyprus, 11–13 December 2013; pp. 109–115. [CrossRef]
12. Barbecho Bautista, P.; Urquiza-Aguiar, L.F.; Aguilar Igartua, M. STGT: SUMO-Based Traffic Mobility Generation Tool for Evaluation of Vehicular Networks. In Proceedings of PE-WASUN 2021, New York, NY, USA, 22–26 November 2021; pp. 17–24. [CrossRef]
13. Fazio, P.; Tropea, M.; Sottile, C.; Lupia, A. Vehicular networking and channel modeling: A new Markovian approach. In Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2015; pp. 702–707. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.