

International Journal of Mathematic Exploration and Computer Education

Cross-Simulator Parity Validation of Quantum Network Entanglement Swapping: A Pre-Declared Validation Methodology with Two Independent Discrete-Event Simulators

David Vesterlund

Industrial Research at Vesterlund Ventures WestQuant Open Source Project.

*Correspondence author

David Vesterlund,

Industrial Research at Vesterlund Ventures WestQuant Open Source Project.

ORCID : <https://orcid.org/0009-0000-6455-1141>

Submitted: 8 Sept 2026; Published: 5 Oct 2026; Published: 19 Oct 2026

Citation: Vesterlund, D. (2026). Cross-Simulator Parity Validation of Quantum Network Entanglement Swapping: A Pre-Declared Validation Methodology with Two Independent Discrete-Event Simulators. *Int J Math Expl & Comp Edu.*3(3):1-8.

Abstract

We report three principal findings. First, two independent discrete-event quantum-network simulators—NetSquid 1.1.8 and SeQUeNCe 1.2.0—agree with an analytical Werner-swap oracle and with each other to machine precision in 160/160 state-level cases ($\max |\Delta F| = 8.88 \times 10^{-16}$) and in 50/50 protocol-level cases under matched source, channel, detector, heralding, retry, cutoff, BSM, readout, occupancy, latency, and goodput semantics. Second, when the shared harness is released and each simulator runs its own asynchronous event engine with independent random streams, mean output fidelity can be equivalent while the fidelity distributions differ strongly (Kolmogorov–Smirnov $p \approx 10^{-27}$); a systematic timestamp-replay diagnosis localises this discrepancy to control/timing sampling rather than quantum-kernel disagreement. Third, a lightweight simulator (QMRB v0.2) with an incorrect Werner swap cross-term (/9 instead of /3) passes all counter-level parity checks but fails all 80 repeater fidelity comparisons with a mean error of 0.0235, demonstrating that model-form errors can be invisible at the counter level and silent until a fidelity threshold is crossed. To obtain these results we introduce a layered, pre-declared validation methodology comprising four layers: state-level parity, protocol-level parity, native-event parity, and model-form diagnosis. Each layer has an acceptance criterion frozen before execution. The methodology distinguishes implementation parity from model-form parity and from protocol-level parity, and reports explicit boundaries on what each layer does and does not establish.

Keywords: Quantum Network Simulation, Cross-Simulator Validation, Entanglement Swapping, Werner States, Model-Form Error, Pre-Declared Criteria.

Introduction

Quantum-network simulation is the primary tool for evaluating routing, scheduling, and entanglement-distribution protocols before hardware deployment [1–7]. Several discrete-event simulators are available—NetSquid, SeQUeNCe, SimQN, QuISP each with different quantum-state formalisms, noise models, event engines, and protocol libraries [1–4]. A result obtained in one simulator is not automatically transferable to another, and the conditions under which simulators agree or disagree are rarely documented [7,8]. A first cross-validation of QuISP and SeQUeNCe reported agreement with each simulator's own analytical model but simulator-specific timing behaviour traceable to different connection and messaging models; to our knowledge no comparable study pairs NetSquid with SeQUeNCe under pre-declared acceptance criteria [8]. This matters because quantum-network control research increasingly uses simulator-specific results to claim algorithmic improvements [9,10]. If two simulators disagree on the same protocol, a routing policy trained in one may not rank the same in the other. If a lightweight analytical model silently uses an incorrect swap formula, downstream policy comparisons

inherit a model-form error that is not visible at the metric level until a fidelity threshold is crossed [12].

We propose a layered validation methodology, in the tradition of simulation verification and validation, and apply it to a controlled Werner-pair protocol [11,12]. The layers are:

1. **State-level Parity:** both simulators receive identical density matrices and storage laws; their quantum kernels must reproduce the analytical oracle.
2. **Protocol-level Parity:** a shared harness supplies identical random numbers and identical source/channel/detector/heralding/retry/cutoff/BSM/readout semantics; only the quantum-state kernel is independently evaluated.
3. **Native-event Parity:** each simulator runs its own asynchronous event engine with independent RNG streams; we test whether rate, fidelity, and latency distributions remain equivalent.
4. **Model-form Diagnosis:** a lightweight simulator is evaluated under the same protocol to detect model-form errors that are invisible at the counter level but visible at the fidelity level.

- Each layer has a pre-declared acceptance criterion frozen before execution, following the logic of preregistration [13]. We report both passes and failures.

Methods

Simulators

Simulator	Version	Quantum formalism	Event engine	License
NetSquid [1]	1.1.8	Density matrix	Native async	Proprietary (free)
SeQUeNCe [2]	1.2.0	Ket, density, Fock, Bell-diag	Timeline (ps)	BSD-3
QMRB-Fast	0.2	Scalar fidelity	Analytical	Internal

NetSquid 1.1.8 is installed from the authenticated NetSquid package index. SeQUeNCe

1.2.0 is installed from PyPI (sequence==1.2.0). Both run on Python 3.12 with NumPy 2.3.5 [14]. QMRB-Fast v0.2 is a scalar-fidelity literature-calibrated package used as a lightweight comparison. All experiments record git commit, Python version, package versions, configuration hash, seeds, and SHA-256 artifact hashes in a manifest.

Pre-Declared Criteria

Acceptance criteria are frozen in a criteria file before any execution [13]. Criteria may not be changed after inspecting results; any revision creates a new version and requires rerunning the affected block.

Layer 1: State-Level Parity

Protocol: Elementary Werner Bell pairs [15] with fidelity $F \in \{0.80, 0.90, 0.95, 0.99\}$. Storage: exponential decay toward the maximally mixed state (fidelity floor $1/4$) [16], parameterised by dimensionless exposure $t/T \in \{0, 0.1, 0.5, 1.0\}$. Swap: ideal Bell-state 2 measurement with outcome-conditioned Pauli correction [17]. Analytical swap oracle [18–20]:

$$F = F F + (1-F)(1-F) / 3 \text{ swap } 1 \ 2 \ 1 \ 2$$

Matrix: 2 topologies $\times$ 4 fidelities $\times$ 4 storage ratios $\times$ 5 seeds $\times$ 100 samples = 160 cases.

Acceptance. Maximum absolute fidelity discrepancy $\leq 10^{-10}$ against the analytical oracle and between simulators.

Layer 2: Protocol-Level Parity

Protocol: A shared harness supplies identical common random numbers and identical definitions for: link success probability ($p \times \text{transmission} \times \eta^2$), source det one attempt per free elementary link per heralding round, memory occupancy (two slots per successful link), cutoff (age $>$ cutoff $\rightarrow$ discard), swap scheduling (attempt when both links ready), BSM success sampling, Werner storage/swap protocol, readout (η^2), delivery (output fidelity $\geq F$), occupancy integration (slot-seconds / ($4 \times \text{readout min duration}$)), goodput (delivered pairs / duration), and latency (older elementary-pair age + readout latency) [21–24].

Matrix: 5 regimes $\times$ 5 seeds = 25 protocol cells, 50 reference-backend rows. **Regimes.**

- metro/default,
- lossy source/channel/detector,
- short memory with tighter cutoff,

- high fidelity threshold,

- slow heralding relative to coherence.

Acceptance. Exact equality for counters; absolute tolerance 10^{-10} for resource/fidelity metrics.

Layer 3: Native-Event Parity

Protocol: Each simulator runs its own asynchronous event engine with independent RNG streams. No common random numbers are shared. Simplified source, channel, detector, memory, BSM, cutoff, readout, and classical-delay controllers are implemented in each simulator's native API.

Matrix: 5 regimes $\times$ 200 seeds per backend. Rate equivalence **margin:** 2% relative. Meanfidelity equivalence margin: 0.005 absolute. TOST [25, 26] with 90% Welch confidence interval [27]. Fidelity and latency distributions: two-sample KS tests [28] at $\alpha = 0.05$.

Layer 4: Model-form diagnosis

QMRB-Fast v0.2 is evaluated under the same state-level and protocol-level protocols. Its swap formula uses a $/9$ Werner cross-term instead of the correct $/3$ [18–20]. We test whether this is visible at the counter level, the fidelity level, and the delivery level.

Results

State-level parity (Layer 1)

Validation	Passed	Total	Max discrepancy
NetSquid vs analytical oracle	160	160	8.88×10^{-16}
SeQUeNCe vs analytical oracle	160	160	8.88×10^{-16}
NetSquid vs SeQUeNCe	160	160	4.44×10^{-16}
QMRB v0.2 vs oracle	80	160	repeater mean error 0.0235
QMRB v0.2 repeater cases	0	80	all fail parity

Both reference simulators match the analytical oracle to machine precision across all tested fidelities, storage ratios, seeds, and topologies. QMRB v0.2 passes all 80 point-topoint cases (no swap applied) and fails all 80 repeater cases. The error is deterministic: the $/9$ cross-term produces $F = F F + (1-F)(1-F)/9$ instead of the correct $/3$, swap 1212 yielding a constant error of 0.00889 at $F=0.80$ and growing with lower fidelity.

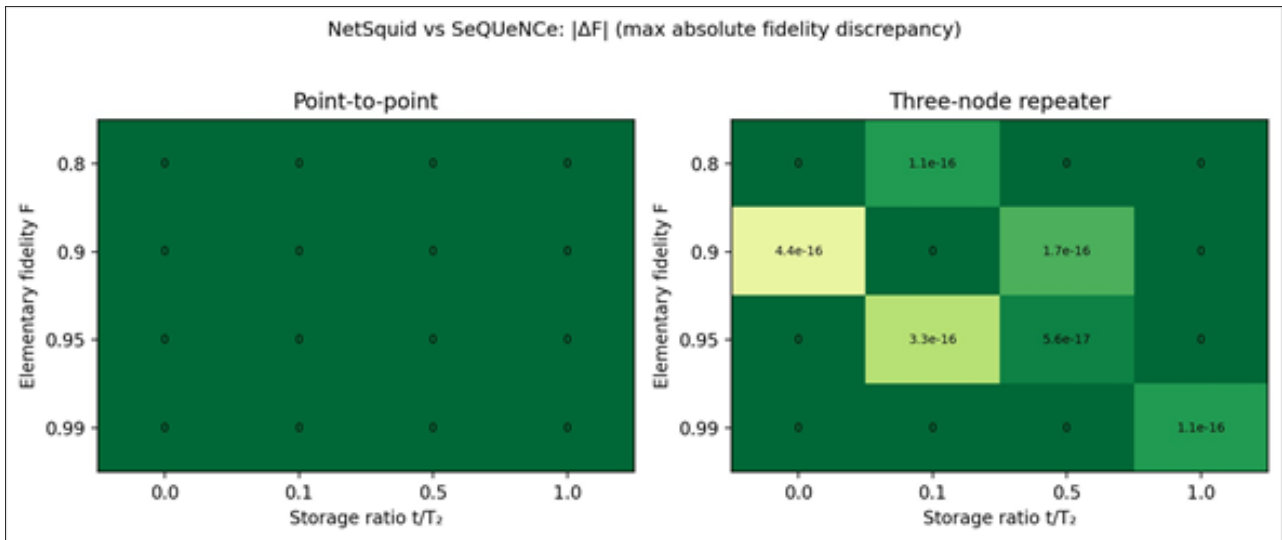


Figure 1: State-level parity: maximum absolute fidelity discrepancy $|\Delta F|$ between NetSquid and SeQUeNCe across all tested fidelities and storage ratios. Both topologies show zero or machine-precision discrepancy

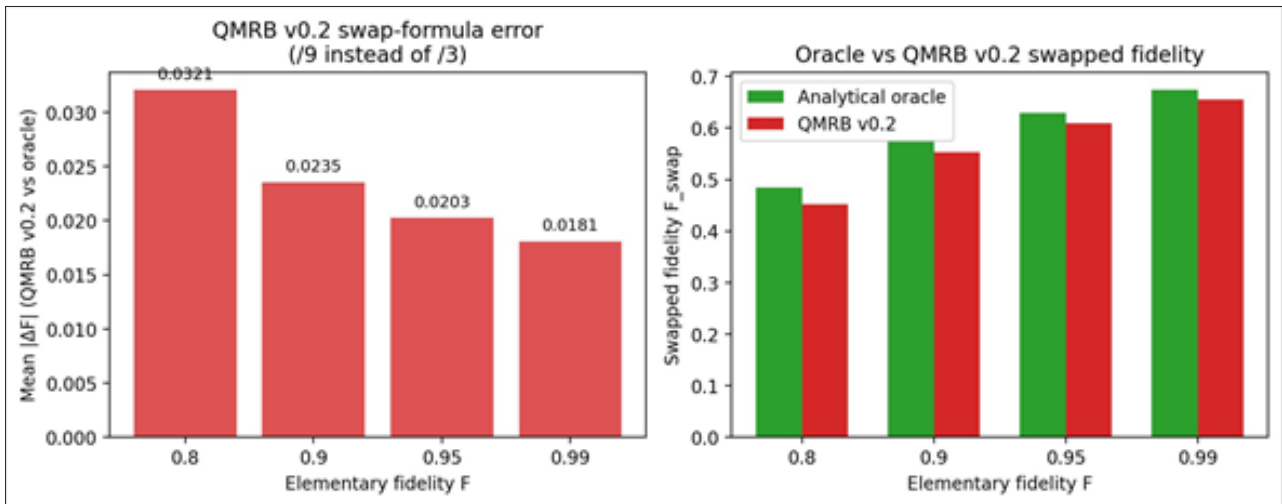


Figure 2: QMRB v0.2 swap-formula error. Left: mean absolute fidelity error by elementary fidelity. Right: oracle vs QMRB swapped fidelity. The /9 cross-term produces a deterministic, growing error

Protocol-level parity (Layer 2)

Criterion	Passed	Total
Source/link/retry/swap/readout/cutoff counters	50	50
Occupancy, slot-seconds, latency	50	50
Delivered-pair and goodput semantics	50	50
Output fidelity (comparable rows)	40	40
Complete matched protocol	50	50

Ten rows have no delivered fidelity to compare (no-delivery cells) and are marked N/A. All 40 comparable fidelity rows pass at tolerance 10^{-10} . QMRB v0.2 under the shared harness: counter parity 25/25, delivery parity 23/25 (two failures in lossy seeds where the swap-map error moves pairs across the fidelity threshold), fidelity parity 0/20 informative comparisons. The five N/A fidelity rows are thresholdstress cells with no delivered fidelity; they are not evidence that the swap map is correct. A corrected QMRB v0.3 (using /3) passes 25/25 delivery and 20/20 informative fidelity comparisons, confirming that the defect is the swap formula, not the harness.

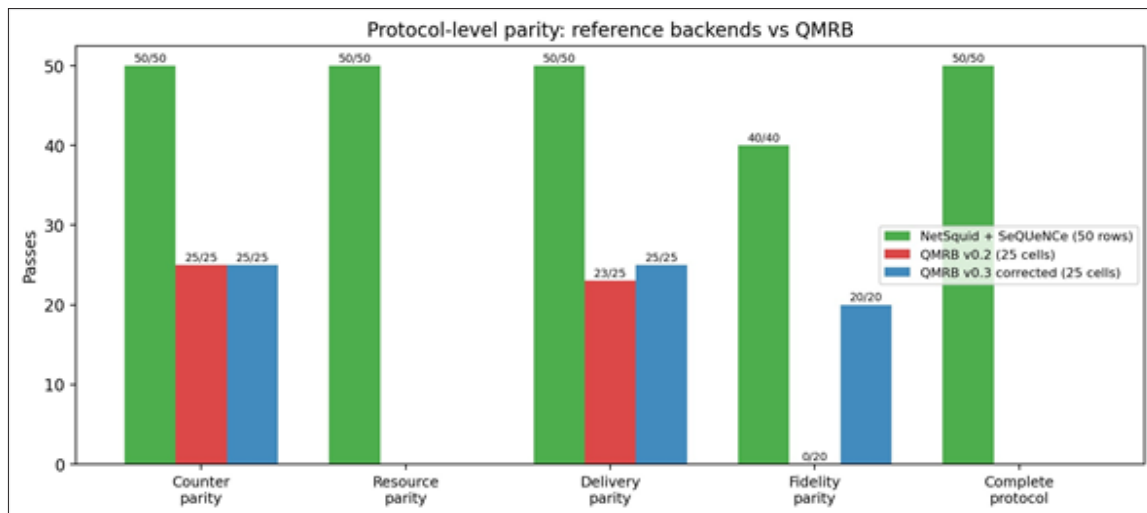


Figure 3: Protocol-level parity summary. NetSquid + SeQuENCe pass all criteria. QMRB v0.2 fails fidelity parity in all informative cases; QMRB v0.3 (corrected /3) restores parity

Native-event parity (Layer 3)

Regime	NS rate	SQ rate	Rate TOST	Fid. TOST	Fid. KS	Lat. KS
Metro	130.75	125.25	Failed	Passed	Failed ($p \approx 10^{-27}$)	Passed
Lossy	13.0	10.0	Failed	Failed	—	—
Short-mem	0	0	N/A	N/A	N/A	N/A
Highthresh	0	0	N/A	N/A	N/A	N/A
Slow-class.	0	0	N/A	N/A	N/A	N/A

Three regimes produced zero deliveries under the short pilot horizon. Rate equality at zero is not evidence of useful equivalence. Key findings:

1. Mean fidelity can be equivalent while fidelity distributions differ strongly.

Metro fidelity TOST passes, but the KS test rejects distribution equality with $p \approx 10^{-27}$.

The distributions have the same mean but different support occupancy.

2. 200 seeds are insufficient for a 2% rate equivalence claim. The metro rate difference CI is $[-6.75, 17.75]$, exceeding the ± 2.56 margin. Failure to establish equivalence is not evidence of a difference [25,26].

3. Sparse regimes produce empty comparisons. Three of five cells delivered zero pairs. These must be redesigned with longer duration or adjusted loads rather than counted as passes.

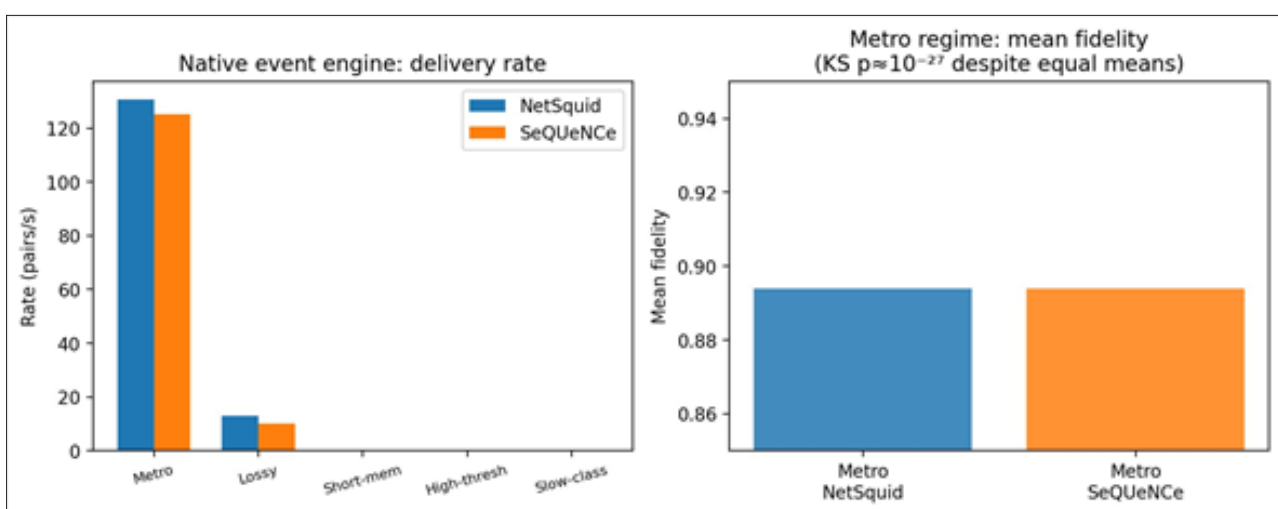


Figure 4: Native-event pilot. Left: delivery rate by regime. Right: metro mean fidelity (equivalent means, but KS rejects distribution equality).

Fidelity diagnosis (Block A)

To localise the metro fidelity-distribution discrepancy, we performed three diagnostic steps:

A1—Delivery Tracing: Across 200 seeds per engine, 1,024 deliveries were recorded. Every delivery contains left/right herald times, swap time, correction arrival, readout time, fidelity, younger/older pair age, age difference, and post-swap wait. No missing trace fields.

Timing diagnosis

Variable	KS D	p-value	NS mean	SQ mean
Younger pair age	0	1.0	0	0
Age difference	0.02093	0.99972	1.0459 ms	1.0279 ms
Post-swap wait	0	1.0	0.2 ms	0.2 ms

None of the tested timing distributions explains the very small fidelity KS p-value. The likely cause is the combination of discrete fidelity support—for which the standard KS null distribution, derived for continuous variables, is not exact [29]—backend-specific independent random-sample occupancy of that support, and an over-sensitive p-value criterion [30]. This supports replacing KS p-value with an effect-size requirement ($D \leq 0.02$). The observed age-difference $D=0.02093$ is itself just above that boundary and should be retested with larger delivery counts.

A2—Timestamp Replay: Each NetSquid delivery timeline was replayed through the SeQUeNCe quantum kernel and vice versa. Maximum absolute fidelity discrepancy: 2.22×10^{-16} against a 10^{-10} criterion.

This confirms the earlier discrepancy is not caused by different quantum-noise evaluation for an identical timeline. It originates in control/timing sampling or distribution-test behaviour.

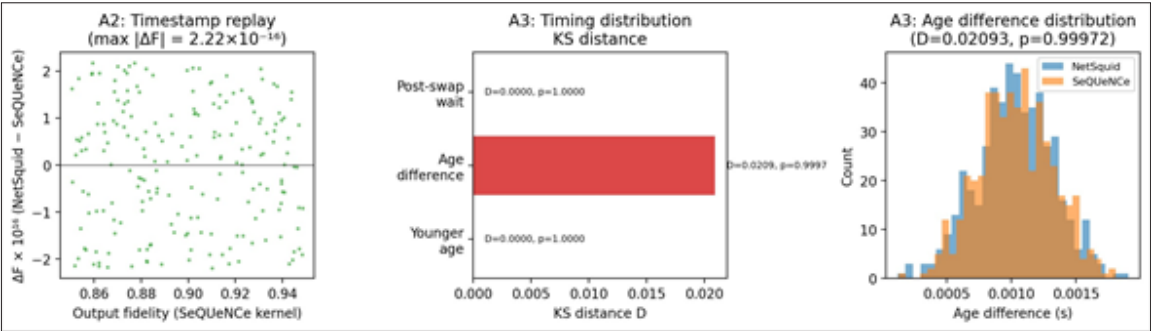


Figure 5: Block A fidelity diagnosis. Left: A2 timestamp replay scatter (max $|\Delta F| = 2.22 \times 10^{-16}$). Centre: A3 timing KS distances. Right: A3 age-difference distributions.

Native noise validation (Block B)

A deterministic native-noise matrix of 4 channels (depolarizing, dephasing, amplitude damping, combined T/T) × 3 state classes (Werner, Bell-diagonal, non-Bell-diagonal) × 12 asymmetric links was evaluated against an independent NumPy [14] Kraus oracle [31].

Metric	Value
Cases	24
Passes	24
Max density delta	5.55×10^{-17}
Max swapped density delta	5.55×10^{-17}
Swap parity passes	24/24

Both simulators' native noise kernels agree with the independent Kraus oracle and with each other to machine precision.

Discussion

What Each Layer Establishes

State-Level Parity: establishes that both simulators implement the same Werner quantum kernel. It does not establish that their event engines, scheduling, or protocol semantics agree.

Protocol-Level Parity: establishes that source, detector, heralding, retry, cutoff, occupancy, latency, delivery, and rate definitions can be held identical. Because scheduling and random draws are shared, counter and resource equality are expected by construction; only the quantum-state kernel is independently evaluated [21]. This is a harness regression gate, not native event-engine parity.

Native-Event Parity: gives the simulators room to disagree. It finds that mean fidelity can be equivalent while distributions differ, and that 200 seeds at short horizon are insufficient for tight rate equivalence. This is the layer where simulator-specific differences become visible, consistent with the timing differences between QuISP and SeQUeNCe reported in [8].

Model-Form Diagnosis: demonstrates that a lightweight simulator can pass counter parity while failing fidelity parity, and that the failure is deterministic and traceable to a single formula error.

Why the Layers are Necessary

A single-layer comparison is insufficient. If we only compared state-level fidelity, we would miss protocol-level disagreements. If we only ran a shared harness, we would miss nativeevent

disagreements. If we only ran native events, we could not distinguish quantumkernel errors from control/timing errors. The layered design localises discrepancies to their source, separating code verification from model validation [11,12].

The QMRB /9 vs /3 Finding

The QMRB v0.2 swap-formula defect is a model-form error: the simulator produces internally consistent counters and correct point-to-point fidelity, but its swap formula uses the wrong Werner cross-term [12]. This is invisible at the counter level and invisible for point-to-point fidelity, but it produces a mean repeater fidelity error of 0.0235 and, near a fidelity threshold, changes delivery outcomes in 2/25 protocol cells. This demonstrates that counter parity is necessary but not sufficient, and that model-form errors can silently affect policy comparisons.

Limitations

- The state-level and protocol-level validations cover a single Werner protocol with ideal BSM and Pauli correction. They do not cover multiplexing, dark counts, detector dead time, purification, heterogeneous memory channels, or arbitrary topology [19,32].
- The native-event pilot uses simplified controllers, not the full built-in hardwarecomponent stacks of either simulator [1,2].
- Three of five native-event regimes produced zero deliveries, yielding empty comparisons.
- The fidelity-distribution discrepancy in the metro regime is not fully explained; the timestamp-replay diagnosis localises it to control/timing sampling, but the specific mechanism requires larger delivery counts and an effect-size criterion.
- No external published target has been reproduced; all comparisons are internal. The entanglement-swapping benchmark of is a natural external target [8].
- The methodology has not been applied to SimQN or QuISP [3,4].

Conclusion

We presented a layered, pre-declared validation methodology for cross-simulator parity in quantum-network simulation. Two independent discrete-event simulators—NetSquid 1.1.8 and SeQUeNCe 1.2.0—agree with an analytical oracle and with each other to machine precision at the state level (160/160) and protocol level (50/50). When allowed to run their own event engines with independent random streams, they exhibit mean-fidelity equivalence but distribution-level disagreement, and 200 seeds are insufficient for a 2% rate-equivalence claim. A systematic timestamp-replay diagnosis localises the distribution discrepancy to control/timing sampling rather than quantum-kernel disagreement. A lightweight simulator with an incorrect swap formula passes counter parity but fails all 80 repeater fidelity comparisons, demonstrating that model-form errors can be invisible at the counter level and silent until a fidelity threshold is crossed. The methodology distinguishes implementation parity from model-form parity and from protocol-level parity, and reports explicit boundaries on

what each layer does and does not establish. We recommend that quantum-network simulation results reported from a single simulator be accompanied by at least state-level parity against an analytical oracle, and that claims involving routing or scheduling policies be validated at the native-event level before being used for cross-simulator transfer claims.

Declaration of Competing Interests

The author declares no competing interests. This research received no external funding. The author has no financial or personal relationships with any of the organizations or products discussed in this work that could inappropriately influence the work presented.

AI Use Disclosure

Generative AI tools (Claude/Anthropic and ChatGPT/OpenAI) were used during the preparation of this manuscript for the following purposes: generating figures and visualizations, drafting and editing prose, formatting the LaTeX manuscript, and assisting with literature search and screening. The research idea, research questions, methodology design, formal mathematical framework, experimental execution, classification of all results, interpretation of findings, and all scientific judgments are solely the work of the author. Every AI-generated figure, table, and text passage has been reviewed, verified, and corrected by the author. The author takes full responsibility for the accuracy and integrity of all content in this manuscript.

References

1. Coopmans, T., et al. (2021). NetSquid, a network simulator for quantum information using discrete events. *Communications Physics*, 4, Article 164. <https://doi.org/10.1038/s42005-021-00647-8>
2. Wu, X., Kolar, A., Chung, J., Jin, D., Zhong, T., Kettimuthu, R., & Suchara, M. (2021). SeQUeNCe: A customizable discrete-event simulator of quantum networks. *Quantum Science and Technology*, 6(4), Article 045027. <https://doi.org/10.1088/2058-9565/ac22f6>
3. Chen, L., Xue, K., Li, J., Yu, N., Li, R., Sun, Q., & Lu, J. (2023). SimQN: A network-layer simulator for the quantum network investigation. *IEEE Network*, 37(5), 182–189.
4. Satoh, R., et al. (2022). QuISP: A quantum internet simulation package. In 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) (pp. 353–364). IEEE. <https://doi.org/10.1109/QCE53715.2022.00056>
5. Wehner, S., Elkouss, D., & Hanson, R. (2018). Quantum internet: A vision for the road ahead. *Science*, 362(6412), Article eaam9288.
6. Azuma, K., et al. (2023). Quantum repeaters: From quantum networks to the quantum internet. *Reviews of Modern Physics*, 95, Article 045006.
7. Bel, O., & Kiran, M. (2025). Simulators for quantum network modelling: A comprehensive review. *Computer Networks*, 263, Article 111204. <https://arxiv.org/abs/2408.11993>

8. Chung, J., Hajdušek, M., Benchasattabuse, N., Kolar, A., Singal, A., Soon, K. S., Teramoto, K., Zang, A., Kettimuthu, R., & Van Meter, R. (2025). Cross-validating quantum network simulators. arXiv. <https://arxiv.org/abs/2504.01290>
9. Pant, M., Krovi, H., Towsley, D., Tassiulas, L., Jiang, L., Basu, P., Englund, D., & Guha, S. (2019). Routing entanglement in the quantum internet. *npj Quantum Information*, 5, Article 25.
10. Shi, S., & Qian, C. (2020). Concurrent entanglement routing for quantum networks: Model and designs. In *Proceedings of the ACM SIGCOMM 2020 Conference* (pp. 62–75). Association for Computing Machinery.
11. Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation*, 7(1), 12–24.
12. Roy, C. J., & Oberkampf, W. L. (2011). A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25–28), 2131–2144.
13. Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11), 2600–2606.
14. Harris, C. R., et al. (2020). Array programming with NumPy. *Nature*, 585, 357–362.
15. Werner, R. F. (1989). Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Physical Review A*, 40, 4277–4281.
16. Razavi, M., Piani, M., & Lütkenhaus, N. (2009). Quantum repeaters with imperfect memories: Cost and scalability. *Physical Review A*, 80, Article 032301.
17. Żukowski, M., Zeilinger, A., Horne, M. A., & Ekert, A. K. (1993). “Event-ready-detectors” Bell experiment via entanglement swapping. *Physical Review Letters*, 71, 4287–4290.
18. Briegel, H.-J., Dür, W., Cirac, J. I., & Zoller, P. (1998). Quantum repeaters: The role of imperfect local operations in quantum communication. *Physical Review Letters*, 81, 5932–5935.
19. Dür, W., Briegel, H.-J., Cirac, J. I., & Zoller, P. (1999). Quantum repeaters based on entanglement purification. *Physical Review A*, 59, 169–181.
20. Munro, W. J., Azuma, K., Tamaki, K., & Nemoto, K. (2015). Inside quantum repeaters. *IEEE Journal of Selected Topics in Quantum Electronics*, 21(3), 78–90.
21. Law, A. M. (2015). *Simulation modeling and analysis* (5th ed.). McGraw-Hill.
22. Sangouard, N., Simon, C., de Riedmatten, H., & Gisin, N. (2011). Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics*, 83, 33–80.
23. Rozpedek, F., et al. (2018). Parameter regimes for a single sequential quantum repeater. *Quantum Science and Technology*, 3(3), Article 034002.
24. Li, B., Coopmans, T., & Elkouss, D. (2021). Efficient optimization of cutoffs in quantum repeater chains. *IEEE Transactions on Quantum Engineering*, 2, Article 4103015.
25. Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6), 657–680.
26. Lakens, D. (2017). Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4), 355–362.
27. Welch, B. L. (1947). The generalization of “Student’s” problem when several different population variances are involved. *Biometrika*, 34(1–2), 28–35.
28. Hodges, J. L. (1958). The significance probability of the Smirnov two-sample test. *Arkiv för Matematik*, 3, 469–486.
29. Conover, W. J. (1972). A Kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, 67(339), 591–596.
30. Wasserstein, R. L., & Lazar, N. A. (2016). The ASA statement on p-values: Context, process, and purpose. *The American Statistician*, 70(2), 129–133.
31. Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th anniversary ed.). Cambridge University Press.
32. Collins, O. A., Jenkins, S. D., Kuzmich, A., & Kennedy, T. A. B. (2007). Multiplexed memory-insensitive quantum repeaters. *Physical Review Letters*, 98, Article 060502.
33. Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10), Article e1003285.

Appendix: Layered Methodology Diagram Reproducibility

All data, code, and artifacts for this paper are available at:

<https://github.com/WestQuantOpen/westquant-qnet>

All experiments record git commit, Python version, package versions, configuration hash, seeds, and SHA-256 artifact hashes, in line with established practice for reproducible computational research [33]. Artifacts are in `results/research_backends/`. The criteria file is frozen at `docs/GATE_D_EXECUTION_CRITERIA.md` before execution.

The repository contains:

- `results/research_backends/` — parity matrices, protocol matrices, native-event pilot data, diagnosis artifacts
- `docs/GATE_D_EXECUTION_CRITERIA.md` — frozen pre-declared criteria
- `scripts/` — all experiment runners and figure-generation scripts
- `src/westquant_qnet/simulators/` — adapter and harness source code

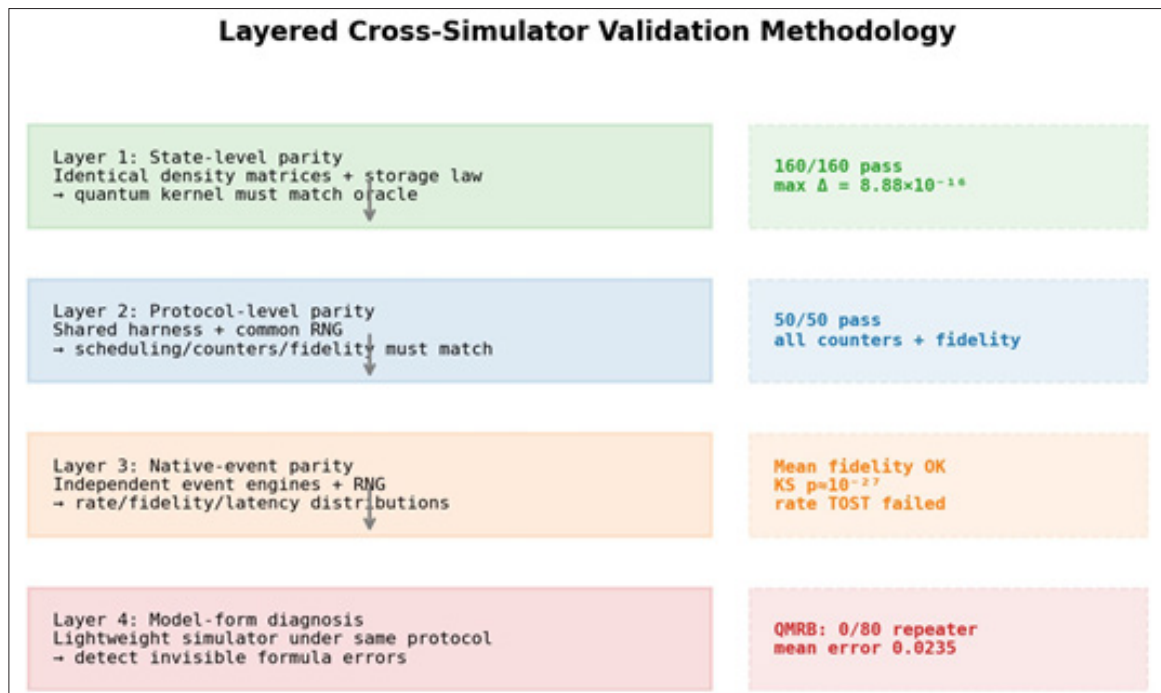


Figure 6: Layered cross-simulator validation methodology. Each layer narrows the scope of what is being tested, from quantum-kernel agreement (Layer 1) to model-form error detection (Layer 4). Arrows indicate dependency: a failure at an earlier layer must be resolved before later layers are interpretable

Copyright: ©2026. David Vesterlund. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.