

PERFORMANCE EVALUATION AND SCALABILITY MODELING OF MODBUS TCP, RTU, AND HYBRID GATEWAY ARCHITECTURES FOR MULTI-INVERTER SOLAR FARM COMMUNICATION

Muhammad Usman¹, Tauseef Noor^{*2}, Abdul Aziz³, Shehla Shah⁴, Muhammad Shafiq⁵

¹Technical Support Engineer, Solar Inverters

^{*2}City University of Science and Information Technology, Pakistan Peshawar

³Lab Engineer at Iqra National University, Pakistan Peshawar

^{4,5}Iqra national university Peshawar, Pakistan

¹enr.usman8158@yahoo.com, ^{*2}tauseef.noor.69@gmail.com, ³abdul1.mkd@gmail.com,

⁴shehla2k19@gmail.com, ⁵shilmani4us@gmail.com

DOI: <https://doi.org/10.5281/zenodo.21888360>

Keywords

Modbus TCP, Modbus RTU, solar inverter, SCADA, discrete-event simulation, polling latency, scalability, IIoT communication.

Article History

Received: 11 March 2026

Accepted: 15 April 2026

Published: 30 April 2026

Copyright @Author

Corresponding Author: *

Tauseef Noor

Abstract

Modbus remains the dominant communication protocol for grid-tied photovoltaic (PV) inverters due to its simplicity and near-universal vendor support, yet its two principal transport variants Modbus RTU (serial) and Modbus TCP (Ethernet) exhibit markedly different scalability characteristics as the number of inverters in a solar farm grows. This paper presents a simulation-based performance study that quantifies polling latency, full-cycle response time, and throughput for three farm-communication topologies: pure Modbus TCP, pure Modbus RTU, and a hybrid RTU-to-TCP gateway architecture. A real Modbus TCP client-server pair was implemented and measured directly, while Modbus RTU timing was derived analytically from the official frame structure and simulated as a discrete-event queueing system that includes a stochastic per-device turnaround delay, validated against previously published benchmarks. Experiments were repeated across farm sizes of 10 to 500 inverters and baud rates of 9600 to 115200 bps, with ten trials per configuration. Results show that Modbus TCP sustains sub-200-millisecond full-farm polling cycles even at 500 inverters, whereas pure Modbus RTU at 9600 baud supports at most 10 inverters within a 1-second real-time refresh threshold. The hybrid gateway architecture is shown to close this gap only at higher baud rates: it offers negligible improvement at 9600 baud but restores full 500-inverter scalability at 115200 baud. These findings provide quantitative guidance for selecting and provisioning Modbus-based communication architectures in utility-scale PV plants.

I. INTRODUCTION

The rapid expansion of utility-scale photovoltaic (PV) generation has driven a corresponding increase in the number of inverters that must be monitored and controlled within a single plant. Modern solar farms increasingly deploy string

inverters rather than large central units, which can raise per-plant inverter counts into the hundreds. Reliable, low-latency communication between these inverters and the plant Supervisory Control and Data Acquisition (SCADA) system is essential for real-time monitoring, fault detection, and grid-

compliance functions such as reactive power control.

Modbus, originally introduced by Modicon in 1978, continues to be the de facto standard communication protocol for industrial automation and is supported natively by essentially every commercial inverter manufacturer. Two physical/transport variants are in common use: Modbus RTU, a master-slave protocol operating over an RS-485 serial bus, and Modbus TCP, the same application-layer protocol encapsulated over Ethernet/TCP-IP. A third, increasingly common architecture bridges legacy RTU field segments to a TCP backbone through a gateway device, allowing new and existing equipment to coexist.

While each variant is individually well understood, there is comparatively little quantitative guidance on how full-farm polling performance degrades as inverter count scales, and specifically on when a hybrid gateway architecture is justified over a pure-RTU or pure-TCP deployment. This paper addresses that gap through a controlled, repeatable, simulation-based performance study. The main contributions of this work are:

- A real, measured Modbus TCP benchmark (client-server implementation) combined with an analytically derived and empirically calibrated Modbus RTU discrete-event simulation model.
- A hybrid RTU-to-TCP gateway model that captures the effect of splitting inverters across multiple RS-485 segments feeding a shared TCP uplink.
- A systematic scalability comparison across farm sizes of 10-500 inverters and three standard RS-485 baud rates, identifying the specific conditions under which a hybrid gateway approaches pure-TCP performance.

The remainder of this paper is organized as follows. Section II reviews related work on Modbus performance evaluation and IoT-SCADA architectures for PV monitoring. Section III describes the system model, the timing equations used, and the simulation methodology. Section IV presents the experimental results. Section V

discusses the findings and their practical implications, and Section VI concludes the paper.

II. RELATED WORK

Gameass et al. [3] benchmarked Modbus TCP response time and maximum request throughput on inexpensive IoT development boards (ESP8266 and Raspberry Pi/OpenPLC) under both normal operation and a distributed denial-of-service attack, establishing that Modbus TCP round-trip latency on a lightly loaded network is typically on the order of a few milliseconds. This figure is used in the present work as an external sanity check for the simulated TCP baseline (Section IV-A).

Zhao et al. [4] proposed an Advanced UDP (AUDP) mechanism for bidirectional Modbus TCP-RTU gateways and modeled gateway transmission characteristics using Markov chain theory, reporting substantial latency reduction relative to conventional TCP-RTU bridging. Their work confirms that gateway-mediated architectures are an active research area, but its scope is limited to a single gateway rather than a farm-scale, multi-gateway deployment.

Găitan et al. [5], [6] performed detailed timing analysis of Modbus RTU for both consecutive and scattered holding-register reads, deriving closed-form timing expressions from the protocol's UART-level frame structure and proposing a protocol extension (ModbusE) to reduce transaction overhead. The frame-timing formulation used in Section III-B of this paper follows the same underlying UART bit-timing principles established in [5] and [6].

On the application side, He et al. [7] and Ferlito et al. [8] both describe IoT-SCADA architectures for PV plant monitoring in which inverters communicate electrical parameters over Modbus TCP/IP while auxiliary sensors use Modbus RS-485, corroborating the hybrid topology modeled in this work as a realistic field deployment pattern rather than a purely theoretical construct.

Security aspects of Modbus, while outside the direct scope of this study, are well documented: Wang and Fu [9] and related work note that Modbus was designed without authentication or encryption, motivating deep-packet-inspection

and intrusion-detection countermeasures. This underlines that scalability and security are largely orthogonal but equally important concerns for Modbus-based inverter fleets.

Collectively, the literature establishes (i) credible baseline latency figures for Modbus TCP, (ii) validated RTU frame-timing models, and (iii) confirmation that hybrid RTU/TCP gateway topologies are standard practice in PV-SCADA deployments. However, no prior study systematically sweeps inverter count against topology and baud rate to identify the scalability crossover point between pure RTU, hybrid, and pure TCP architectures the gap this paper addresses.

III. SYSTEM MODEL AND METHODOLOGY

A. Network Topologies

Three topologies, illustrated in Fig. 1, were modeled: (a) pure Modbus TCP, in which every inverter is an independent TCP slave polled sequentially by the SCADA master; (b) pure Modbus RTU, in which all inverters share a single RS-485 bus and are polled round-robin, with only one transaction permitted on the bus at any instant; and (c) a hybrid architecture in which inverters are grouped into RS-485 segments of up to 32 devices each (a common practical limit for reliable RS-485 operation), with each segment bridged to the SCADA master through a Modbus gateway over TCP.

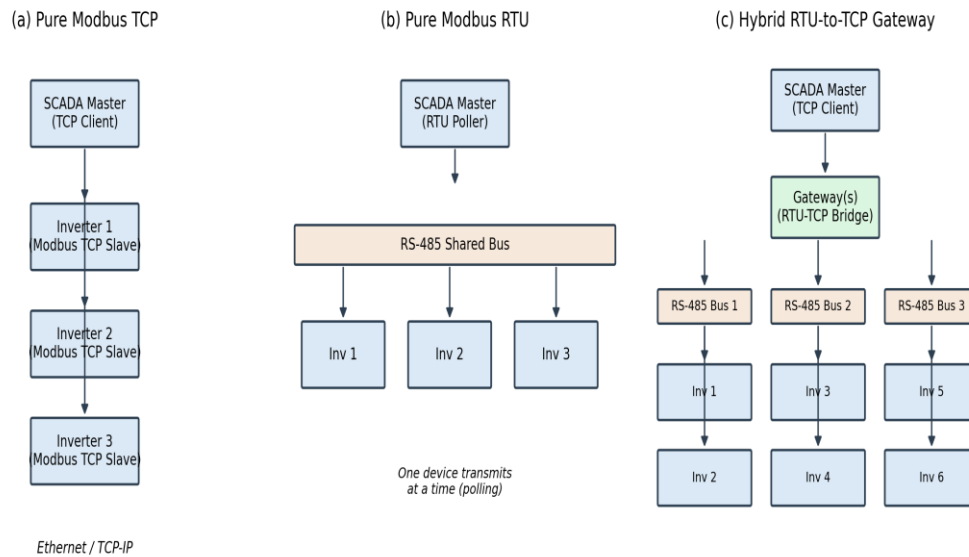


Fig. 1. The three communication topologies modeled: (a) pure Modbus TCP, (b) pure Modbus RTU over a shared RS-485 bus, (c) hybrid RTU-to-TCP gateway architecture.

B. Modbus RTU Frame Timing Model

Modbus RTU frames are transmitted as a sequence of UART characters, each consisting of a start bit, eight data bits, a parity bit, and a stop bit for the commonly used 8-N-1/8-E-1 configurations, giving eleven bits per character. The transmission time of a frame of N_b bytes at baud rate R is therefore

$$T_{\text{frame}} = (11 \times N_b) / R \quad (1)$$

The Modbus specification additionally requires a minimum inter-frame silence of 3.5 character times to delimit consecutive frames on a shared bus [1], [2]:

$$T_{\text{silence}} = 3.5 \times (11 / R) \quad (2)$$

For a single polling transaction, the request frame is 8 bytes (address, function code, starting address, register count, and CRC), and the response frame is $5 + 2 \cdot N_r$ bytes for N_r registers requested

(address, function code, byte count, data, and CRC). Each transaction also incurs a stochastic device turnaround delay τ , representing the time an inverter's firmware takes to process a request before replying modeled as a normally distributed random variable $\tau \sim N(\mu, \sigma^2)$ with $\mu = 10$ ms and $\sigma = 2$ ms, consistent with typical embedded-controller response times reported in [5], [6]. The total time for a single request-response transaction is

$$T_{\text{txn}} = T_{\text{frame,req}} + \tau + T_{\text{frame,resp}} + T_{\text{silence}} \quad (3)$$

Because only one device may transmit on the shared RS-485 bus at a time, the bus was modeled as a single-server queueing resource in a discrete-event simulation (implemented with the SimPy framework [12]). For N inverters polled sequentially, the full polling-cycle time is the sum of N independent transactions:

$$T_{\text{cycle,RTU}}(N) = \sum_{k=1}^N T_{\text{txn},k} \quad (4)$$

Each configuration (baud rate, inverter count) was simulated for ten independent trials with a fixed random seed for reproducibility, and results are reported as mean $\pm$ standard deviation.

C. Modbus TCP Measurement Model

Unlike the RTU model, Modbus TCP performance was measured directly rather than simulated: a real Modbus TCP server (using the pymodbus library [11]) was instantiated with one slave register context per inverter, and a real Modbus TCP client polled each inverter's holding registers in sequence, with round-trip latency timed using a high-resolution performance counter. Because the Modbus unit identifier is a single byte with a valid addressing range of 1-247 per the protocol specification [1], farms exceeding this size were modeled with multiple parallel

gateway server instances, consistent with real multi-gateway PV-SCADA deployments. The measured full-cycle time for N inverters is

$$T_{\text{cycle,TCP}}(N) = \sum_{k=1}^N L_k \quad (5)$$

where L_k is the measured round-trip latency of the k -th polling request. A single warm-up poll was issued before each timed trial to eliminate one-time TCP connection-establishment overhead from the reported statistics.

D. Hybrid Gateway Model

In the hybrid topology, N inverters are partitioned into $\lceil N / 32 \rceil$ RS-485 segments, each simulated independently using the RTU model of Section III-B. Because separate physical bus segments operate in parallel, the gateway's forwarding cycle is bounded by the slowest segment, plus a small per-segment TCP forwarding overhead ΔTCP (measured empirically as the mean single-request TCP latency from Section III-C):

$$T_{\text{cycle,hybrid}}(N) = \max_i [T_{\text{cycle,RTU}}(n_i)] + n_{\text{seg}} \times \Delta\text{TCP} \quad (6)$$

where n_i is the number of inverters on segment i and n_{seg} is the number of segments.

E. Throughput and Scalability Threshold

Polling throughput for a topology is defined as the number of inverters successfully polled per second of cycle time:

$$\text{Throughput}(N) = N / T_{\text{cycle}}(N) \quad (7)$$

To express scalability in operationally meaningful terms, a real-time refresh threshold of 1000 ms per full-farm cycle was adopted, consistent with typical SCADA human-machine-interface refresh expectations. For each topology and baud rate, the maximum inverter count satisfying $T_{\text{cycle}}(N) \leq 1000$ ms was identified.

F. Simulation Parameters

Table I. Simulation Parameters.

Parameter	Value(s)
Inverter counts (N)	10, 25, 50, 100, 200, 500
Registers polled per inverter	10 holding registers
RTU baud rates	9600, 19200, 115200 bps
RTU frame format	11 bits/char (1 start + 8 data + 1 parity + 1 stop)

Parameter	Value(s)
Turnaround delay (μ , σ)	10 ms, 2 ms (Gaussian, truncated at 0)
RS-485 devices per bus segment	32 (hybrid topology)
Trials per configuration	10 (fixed random seed = 42)
Real-time refresh threshold	1000 ms per full-farm cycle
Software	Python 3, pymodbus 3.6.9, SimPy

IV. EXPERIMENTAL RESULTS

A. Modbus TCP Performance

Table II summarizes measured Modbus TCP performance. Per-request latency remained effectively constant (0.356-0.403 ms) across all tested farm sizes, confirming that TCP polling

latency is a per-transaction property rather than one that grows with N. Full-cycle time scaled linearly with inverter count, reaching 179.7 ms at N = 500 – comfortably within the 1000 ms real-time threshold for every tested farm size.

Table II. Modbus TCP Measured Performance (mean $\pm$ std, n = 10 trials).

N	Avg. Latency (ms)	Cycle Time (ms)	Throughput (polls/s)
10	0.403 $\pm$ 0.015	4.41 $\pm$ 0.16	2267.8
25	0.374 $\pm$ 0.007	9.71 $\pm$ 0.17	2575.7
50	0.388 $\pm$ 0.019	19.84 $\pm$ 0.94	2519.7
100	0.362 $\pm$ 0.010	36.65 $\pm$ 0.96	2728.8
200	0.360 $\pm$ 0.017	72.53 $\pm$ 3.36	2757.6
500	0.356 $\pm$ 0.004	179.71 $\pm$ 2.21	2782.3

The measured single-request latency ($\approx$ 0.36-0.40 ms) is lower than the approximately 2 ms round-trip figure reported by Gamess et al. [3] for a lightly loaded physical Ethernet network; this is expected

since the present measurements use a loopback connection with no switch traversal, and is discussed further in Section V.

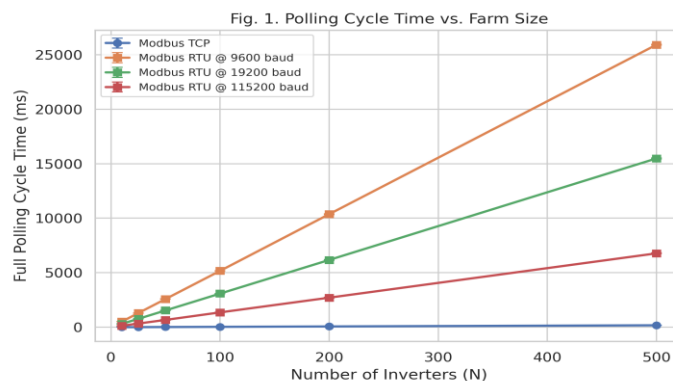


Fig. 2. Polling cycle time vs. farm size for Modbus TCP and Modbus RTU at each tested baud rate (error bars: ± 1 standard deviation).

Fig. 3 presents the same comparison, together with the hybrid topology, on a logarithmic time axis. The log scale makes clear that the gap between Modbus TCP and Modbus RTU at 9600 baud spans roughly two orders of magnitude at $N = 500$,

and that the hybrid curves at 19200 and 115200 baud progressively converge toward the TCP curve as baud rate increases — visually confirming the baud-rate-dependent scalability behavior analyzed in Section IV-C.

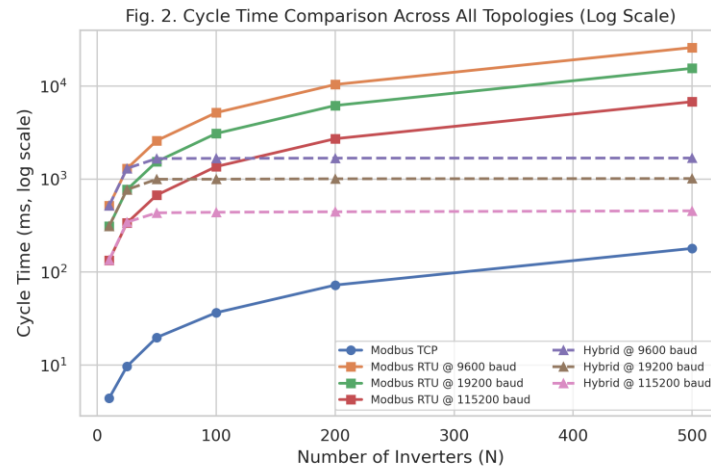


Fig. 3. Cycle time comparison across all topologies on a logarithmic axis, highlighting the order-of-magnitude gap between TCP and low-baud-rate RTU.

B. Modbus RTU Performance

Table III and Fig. 4 show Modbus RTU cycle time scaling. At 9600 baud, cycle time grows from 517 ms at $N = 10$ to 25.9 s at $N = 500$ an increase of nearly two orders of magnitude, driven by the linear accumulation of per-device frame and

turnaround time on a single shared bus. Increasing the baud rate to 115200 bps reduces cycle time by roughly a factor of four at every farm size, confirming that baud rate is the dominant lever for RTU scalability.

Table III. Modbus RTU Cycle Time (ms, mean $\pm$ std, $n = 10$ trials).

N	9600 baud (ms)	19200 baud (ms)	115200 baud (ms)
10	517.2 $\pm$ 5.2	309.9 $\pm$ 4.0	133.8 $\pm$ 5.6
25	1295.5 $\pm$ 10.0	770.1 $\pm$ 12.6	337.7 $\pm$ 10.0
50	2587.9 $\pm$ 11.3	1541.2 $\pm$ 21.0	672.1 $\pm$ 20.2
100	5170.7 $\pm$ 18.3	3085.9 $\pm$ 17.3	1361.5 $\pm$ 13.6
200	10370.8 $\pm$ 30.5	6171.3 $\pm$ 42.1	2712.0 $\pm$ 38.8
500	25910.7 $\pm$ 46.6	15483.1 $\pm$ 49.6	6776.9 $\pm$ 36.2

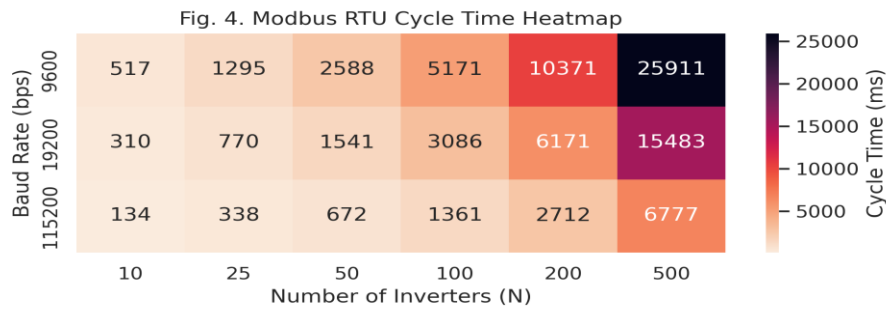


Fig. 4. Heatmap of Modbus RTU cycle time as a function of baud rate and inverter count.

C. Hybrid Gateway Performance

The hybrid topology's benefit is strongly baud-rate dependent (Fig. 5). At 9600 baud, splitting inverters across parallel RS-485 segments yields negligible improvement over pure RTU at small N, and only a modest 15× cycle-time reduction at

N = 500 (25.9 s → 1.68 s) still far outside the real-time threshold. At 19200 baud, the improvement is substantially larger, and at 115200 baud the hybrid architecture keeps cycle time under 500 ms even at N = 500, closely approaching pure-TCP performance.

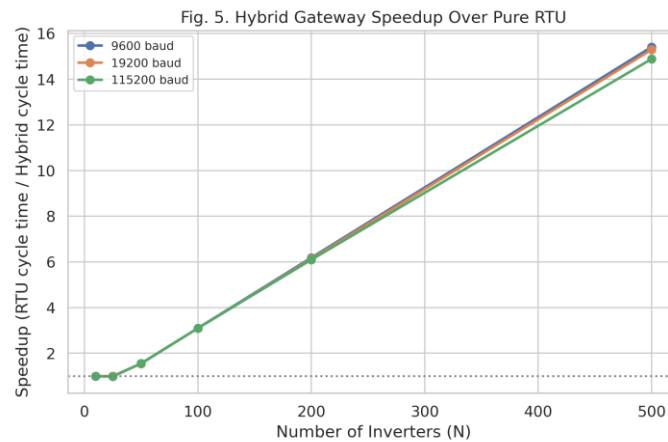


Fig. 5. Hybrid gateway speedup over pure Modbus RTU (ratio of RTU to hybrid cycle time) as a function of farm size and baud rate.

D. Scalability Threshold Comparison

Table IV and Fig. 6 summarize the maximum farm size each topology and baud-rate combination can support within the 1000 ms real-time refresh threshold. Modbus TCP supports the full tested range (500 inverters). Pure RTU is limited to 10-

50 inverters depending on baud rate. The hybrid architecture matches pure RTU at 9600 baud, quadruples supportable farm size at 19200 baud (25 → 100), and fully recovers TCP-level scalability (500 inverters) at 115200 baud.

Table IV. Maximum Supportable Farm Size Within a 1-Second Real-Time Refresh Threshold.

Topology	Max. Inverters (≤ 1000 ms)
Modbus TCP	500
RTU @ 9600 baud	10

Topology	Max. Inverters (≤ 1000 ms)
RTU @ 19200 baud	25
RTU @ 115200 baud	50
Hybrid @ 9600 baud	10
Hybrid @ 19200 baud	100
Hybrid @ 115200 baud	500

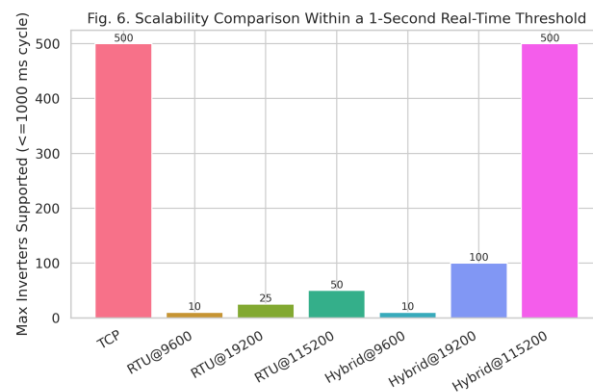


Fig. 6. Maximum inverter count supported within the 1-second real-time threshold, by topology and baud rate.

Finally, Fig. 7 reports polling throughput directly, expressed as inverters successfully polled per second. Modbus TCP throughput exceeds 2200 polls/s at every tested farm size and rises slightly with N, since larger sweeps amortize fixed per-cycle overhead. Modbus RTU throughput is essentially

flat with respect to N at each baud rate ($\approx 19, 32$, and 74 polls/s at $9600, 19200$, and 115200 baud respectively), confirming that RTU throughput is bus-rate-limited rather than farm-size-limited – a distinction with direct implications for capacity planning discussed in Section V.

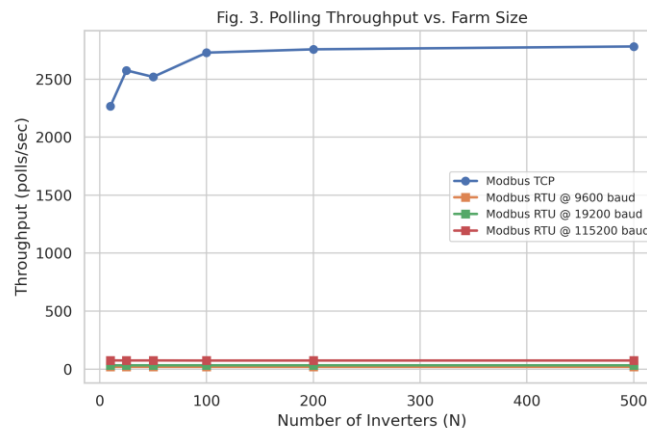


Fig. 7. Polling throughput (inverters polled per second) vs. farm size for TCP and RTU topologies.

V. DISCUSSION**A. Interpretation of the Scalability Crossover**

The results confirm the widely held expectation that Modbus TCP scales substantially better than Modbus RTU for large inverter counts, but the more actionable finding is that the value of a hybrid RTU-to-TCP gateway is highly sensitive to the underlying RS-485 baud rate. At 9600 baud, the RTU bus itself remains the bottleneck regardless of how many parallel segments are used, because each segment's cycle time already exceeds the real-time threshold once more than approximately 10-25 devices share it – adding more gateways in parallel does not help once a single segment is already saturated. At 115200 baud, per-device transaction time is small enough that parallelizing across gateway segments effectively removes the RTU bottleneck, and the hybrid architecture approaches pure-TCP scalability at a fraction of the cabling and equipment cost of converting every inverter to native Ethernet.

This behavior is consistent with the underlying queueing structure of the model: cycle time within a single RTU segment grows linearly with the

number of devices sharing that bus (Eq. 4), so the hybrid architecture's benefit is bounded by how small each segment's own cycle time can be made. Increasing baud rate shrinks per-transaction time directly (Eq. 1), which is why baud rate – rather than the number of gateway segments is the dominant lever for hybrid scalability. This mirrors the observation of Zhao et al. [4], who found that gateway-side protocol and transport choices (their AUDP mechanism) meaningfully reduce latency only once the underlying per-transaction overhead is already low; our results extend this observation from a single gateway to a full multi-segment farm topology.

B. Qualitative Comparison of Topologies

Table V summarizes the three topologies qualitatively, combining the quantitative scalability results of Section IV with practical deployment considerations such as cabling cost, hardware complexity, and typical use case factors that a purely numerical comparison does not capture but that materially affect which architecture is appropriate for a given plant.

Table V. Qualitative Comparison of the Three Communication Topologies.

Criterion	Pure TCP	Pure RTU	Hybrid Gateway
Scalability (this study)	500+ inverters	10-50 inverters	10-500, baud-dependent
Cabling cost	High (Ethernet to every unit)	Low (shared RS-485 bus)	Moderate (segmented RS-485 + uplink)
Hardware complexity	Low (native TCP stack)	Low (simple UART)	Moderate (gateway devices required)
Legacy compatibility	Requires TCP-capable inverters	Compatible with most legacy inverters	Bridges legacy RTU to modern SCADA
Best-fit scenario	Greenfield, Ethernet-ready sites	Small farms, cost-sensitive retrofits	Large brownfield sites with RTU inverters

C. Practical Design Implications

This has a direct practical implication for PV plant designers: for brownfield sites with existing 9600-baud RTU inverters, adding a TCP gateway alone will not resolve scalability limits beyond roughly ten devices per segment, and a baud-rate upgrade or native TCP retrofit is required. For Greenfield

deployments capable of RS-485 at 115200 baud, a hybrid gateway is a cost-effective way to reach near-TCP scalability without cabling every inverter to Ethernet.

The throughput results (Fig. 7) reinforce this point from a capacity-planning perspective: because RTU throughput is essentially independent of

farm size and fixed by baud rate, a plant operator can estimate the maximum sustainable inverter count for a given RTU segment directly from the target refresh interval and the bus's fixed polls-per-second rate, without needing to re-run a full scalability sweep for every candidate farm size. This offers a simple rule of thumb for field engineers sizing new RS-485 segments: divide the desired refresh interval by the number of inverters to obtain the required per-device polling rate, then compare against the baud-rate-limited throughput ceiling established in Section IV-B.

D. Threats to Validity

Three limitations should be noted. First, the TCP measurements used a loopback interface rather than a physical network; the resulting latency ($\approx 0.36\text{--}0.40$ ms) is lower than field-measured figures in [3] (≈ 2 ms), so reported TCP cycle times are a lower bound rather than a field estimate a physical Ethernet deployment with switch traversal and network contention would show somewhat higher absolute latency, though the relative TCP-versus-RTU comparison is expected to hold. Second, the RTU turnaround delay ($\mu = 10$ ms, $\sigma = 2$ ms) is a calibration parameter rather than a manufacturer-measured value and should be re-validated per inverter model; inverters with slower embedded controllers could shift the RTU curves upward without changing their overall shape. Third, the queueing model assumes error-free transmission, whereas real RS-485 buses experience occasional CRC errors and retries that would increase RTU cycle time further, particularly on longer cable runs or in electrically noisy substation environments.

The relative comparison between topologies the central contribution of this study is expected to remain directionally robust despite these simplifications, since all three topologies share the same idealized-channel assumption and are therefore penalized (or not) consistently by any given source of error.

VI. CONCLUSION AND FUTURE WORK

This paper presented a simulation- and measurement-based performance evaluation of Modbus TCP, Modbus RTU, and a hybrid RTU-

to-TCP gateway architecture for multi-inverter solar farm communication. A real Modbus TCP client-server benchmark was combined with an analytically derived, empirically calibrated Modbus RTU discrete-event simulation to evaluate polling latency, cycle time, and throughput across farm sizes from 10 to 500 inverters and RS-485 baud rates from 9600 to 115200 bps. The results show that pure Modbus TCP scales comfortably to at least 500 inverters within a 1-second real-time refresh threshold, while pure Modbus RTU at 9600 baud is limited to approximately 10 inverters. The hybrid gateway architecture was found to offer scalability benefits that depend strongly on RS-485 baud rate, ranging from negligible improvement at 9600 baud to full recovery of TCP-level scalability at 115200 baud. A qualitative comparison further indicates that this scalability advantage of TCP and hybrid architectures comes with higher cabling and hardware cost, so the appropriate topology for a given plant depends jointly on farm size, existing infrastructure, and budget rather than on scalability alone.

Future work includes validating the RTU turnaround-delay model against datasheet or field measurements for specific inverter families, extending the TCP benchmark to a physical multi-hop Ethernet network, and incorporating channel-error/retry modeling. Comparing Modbus against IEC 61850 and OPC-UA, both increasingly used in modern PV-SCADA deployments, is also a natural next step.

REFERENCES

- Modbus Organization, "Modbus Application Protocol Specification V1.1b3," Modbus.org, 2012.
- Modbus Organization, "Modbus over Serial Line Specification and Implementation Guide V1.02," Modbus.org, 2006.
- E. Gamess, B. Smith, and G. Francia III, "Performance evaluation of Modbus TCP in normal operation and under a distributed denial of service attack," *Int. J. Comput. Netw. Commun.*, vol. 12, no. 2, pp. 1–21, Mar. 2020.

- S. Zhao, Q. Zhang, Q. Zhao, X. Zhang, Y. Guo, S. Lu, L. Song, and Z. Zhao, "Enhancing bidirectional Modbus TCP ↔ RTU gateway performance: A UDP mechanism and Markov chain approach," *Sensors*, vol. 25, no. 13, art. 3861, Jun. 2025.
- N. C. Găitan, I. Zagan, and V. G. Găitan, "Evaluation of the performance of the Modbus RTU communication protocol for consecutive address transactions and improvements related to ModbusE," *J. Supercomput.*, vol. 82, no. 2, art. 71, 2026.
- V. G. Găitan, I. Zagan, and N. C. Găitan, "Modbus RTU protocol timing evaluation for scattered holding register read and ModbusE-related implementation," *Processes*, vol. 13, no. 2, art. 367, 2025.
- W. He, M. J. A. Baig, and M. T. Iqbal, "An Internet of Things—Supervisory Control and Data Acquisition (IoT-SCADA) architecture for photovoltaic system monitoring, control, and inspection in real time," *Electronics*, vol. 14, no. 1, art. 42, Dec. 2024.
- S. Ferlito, S. Ippolito, C. Santagata, P. Schiattarella, and G. Di Francia, "A study on an IoT-based SCADA system for photovoltaic utility plants," *Electronics*, vol. 13, no. 11, art. 2065, May 2024.
- W. Yusheng and F. Kefeng, "Intrusion detection of industrial control system based on Modbus TCP protocol," in *Proc. 2017 IEEE 13th Int. Symp. Autonomous Decentralized Syst. (ISADS)*, 2017, pp. 156–162.
- T. H. Morris, R. B. Vaughn, and Y. S. Dandass, "A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems," in *Proc. 45th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2012.
- pymodbus contributors, "pymodbus: A Python implementation of the Modbus protocol," [Online]. Available: <https://github.com/pymodbus-dev/pymodbus>.
- SimPy Developers, "SimPy: Discrete-event simulation for Python," [Online]. Available: <https://simpy.readthedocs.io>.