# ENERGY OPTIMIZATION OF NETWORKS THROUGH A HYBRID SDN-CRDT ARCHITECTURE: AN EXPLORATORY STUDY BASED ON SYNTHETIC DATA.


## Abstract

Increased traffic, denser infrastructure, and the proliferation of distributed services are reinforcing the need for network management strategies that balance performance and energy efficiency. Software-Defined Networking (SDN) simplifies network programming by decoupling the control plane from the forwarding plane, while Conflict-Free Replicated Data Types (CRDTs) enable the convergence of replicated data without requiring synchronous coordination for each update. This article explores the potential benefits of a hybrid SDN–CRDT architecture for distributed control state management. A synthetic dataset of 270 observations was generated for three architectures (conventional network, SDN alone, and SDN–CRDT), three network sizes (20, 50, and 100 nodes), three traffic loads, and ten hypothetical repetitions per scenario. The indicators include one-hour energy consumption, latency, throughput, convergence time, control overhead, CPU utilization, and packet loss. At medium load, the hybrid model's synthetic power consumption is 3.3%, 5.8%, and 10.3% lower than that of the conventional network for 20, 50, and 100 nodes, respectively. However, the model attributes these trends to the chosen generation equations; they do not demonstrate a measurable gain or a true superiority. The article formalizes the assumptions, outlines the limitations, and proposes a reproducible validation protocol under Mininet or NS-3.



Keywords: Software-Defined Networking; CRDT; energy efficiency; distributed networks; convergence; simulation; state replication.


## 1. Introduction

Modern networks must support increasing connectivity demands, applications with heterogeneous requirements, and constantly evolving topologies. Enterprise infrastructures, data centers, access networks, the Internet of Things (IoT), and edge computing architectures rely on switching, routing, computing, and control equipment whose energy consumption depends on load, installed capacity, and management decisions. Energy efficiency cannot therefore be assessed solely based on the instantaneous power consumption of a piece of equipment; it must also consider the volume of traffic carried, quality of service, availability, and the cost of control mechanisms.

In a traditional approach, control decisions are generally distributed across equipment and implemented using protocols and configurations specific to each domain. This distribution can offer local autonomy,

UNDER PEER REVIEW IN IJAR

but it sometimes makes overall optimization difficult: independently made decisions do not necessarily lead to coordinated resource allocation. Energy-saving strategies can include path adaptation, putting underutilized resources into standby mode, positioning forwarding rules, and managing link activation rates. These decisions are subject to a trade-off between power consumption, latency, available capacity, and resilience. The literature on energy efficiency in SDN emphasizes the importance of traffic-sensitive management, endpoint considerations, and rule placement, while noting that the gains depend on the scenario and service constraints (Assefa & Özkasap, 2019).

Software-Defined Networking (SDN) offers a programmable abstraction that separates the control plane from the data plane, allowing management applications to control network behavior through defined interfaces. This separation facilitates telemetry collection, the implementation of cross-cutting policies, and the adaptation of forwarding rules. The synthesis by Kreutz et al. (2015) presents SDN as a flexible paradigm, but also raises issues related to security, controller availability, scalability, and decision consistency. Therefore, SDN does not automatically guarantee energy reduction: a central controller or a set of controllers can generate a computational and communication load that must be factored into the assessment.

Replicating the control state across multiple instances can contribute to availability and service continuity, but it introduces a consistency problem. A solution based on strict coordination may require synchronous exchanges and increase latency, particularly when controllers are geographically distributed or the network is partitioned. Conflict-Free Replicated Data Types (CRDTs) are data structures designed to allow multiple replicas to process concurrent updates and converge once the necessary updates have been propagated. The framework formalized by Shapiro et al. (2011) establishes convergence conditions for CRDTs, while delta-state CRDTs potentially reduce replication volume by transmitting fragments of change rather than the complete state (Almeida et al., 2018). These properties relate to data consistency; they do not, in themselves, constitute an energy-saving mechanism.

The combination of SDN and CRDT thus represents an architectural avenue worth exploring, provided that the replicated information, permitted operations, and the consequences of delayed convergence on routing decisions are precisely defined. Lighter replication could reduce certain synchronization exchanges or accelerate recovery after a loss of connectivity between controllers. Conversely, replica maintenance, update propagation, metadata management, and merge operations consume resources. The energy balance therefore depends on the ratio between the cost of replication and the savings achieved through improved network coordination.

This work aims to construct an exploratory framework for comparing three architectures: (i) a classic network architecture, (ii) a replication-free SDN with CRDT, and (iii) a hybrid SDN-CRDT architecture.

UNDER PEER REVIEW IN IJAR

The research question is: under what conditions could distributed control state management using CRDT be associated with lower energy consumption without excessively degrading latency, throughput, or reliability? The immediate objective is not to claim experimental validation, but rather to define the variables to be observed, illustrate a comparison pipeline, and establish a reproducible evaluation protocol.

This manuscript makes four contributions. First, it presents a conceptual architecture in which CRDTs replicate a subset of the SDN control state. Second, it describes a synthetic dataset of 270 observations to test the structure of the analyses and visualizations. Third, it compares the trends produced by this model according to network size and load. Fourth, it clearly distinguishes results generated by assumptions from conclusions that would require real-world measurement, and proposes a validation strategy in a network simulation environment.

## 2. Materials and methods

### 2.1. Architectural framework

The conceptual architecture comprises three functional layers. The data layer includes the switches and links that ensure packet forwarding. The control layer includes one or more SDN controllers responsible for calculating or distributing forwarding rules. The management and optimization layer leverages traffic metrics, quality of service indicators, and energy policies to propose configuration changes. This decomposition is consistent with the general SDN terminology described in IETF/IRTF RFC 7426, which notably distinguishes between forwarding, control, and management functions (Haleplidis et al., 2015).

In the classic SDN variant considered here, a controller or set of controllers manages the network state according to the chosen implementation mechanism. In the hybrid variant, certain control data is replicated using a CRDT (Controller-Remote Data Transfer) between multiple control instances. Candidate data includes load counters, link observation information, path calculation requests, and monitoring metadata. However, operations that must adhere to security or capacity constraints should not be automatically made contingent without analysis. For example, a link activation/deactivation decision can only be applied after verifying connectivity, capacity, and loop protection constraints.

The CRDT should be considered a mechanism for converging the replicated state, not a routing controller. The decision logic remains with the SDN application. A filtering mechanism must distinguish commutative data, which can be merged without strict coordination, from operations dependent on a global order or invariants. This separation limits the risk that correct convergence at the data level might nevertheless produce an incorrect network configuration.

### 2.2. Proposed CRDT Mechanism

UNDER PEER REVIEW IN IJAR

CRDTs are generally divided into state-based and operation-based structures. In state-based CRDTs, replicas merge states using a join function that satisfies the mathematical properties necessary for convergence. In operation-based CRDTs, operations are propagated and applied to other replicas according to specific diffusion assumptions. Delta-state CRDTs transmit state increments, which can reduce the communication cost when the changes are small compared to the complete state (Almeida et al., 2018).

For a prototype, a distributed counter like a G-Counter could aggregate monotonic events, such as the number of alerts or observations, while a register or ensemble-type structure could represent certain monitoring states. The final choice should depend on the data semantics. Load counters, for example, must be associated with a time window and an expiration policy: the convergence of monotonic values does not mean that an old value remains relevant for routing decisions. A production solution should also handle timestamps, duplicates, stale replicas, member discovery, authentication, and metadata cleansing.

The expected benefit of CRDT replication is therefore conditional. It could materialize if replication avoids blocking synchronizations, improves the availability of control data, or reduces the amount of data exchanged through deltas. The additional cost includes update transmissions, merges, metadata storage, and the consumption of control nodes. These costs must be measured separately to identify the break-even point.

### 2.3. Design of the synthetic dataset

A synthetic dataset was generated to illustrate the comparative treatment and verify the structure of the tables and figures. It comprises 270 observations, resulting from the combination of three architectures, three network sizes (20, 50, and 100 nodes), three load levels (low, medium, and high), and ten hypothetical replicates per combination. The load levels are represented by factors of 0.35, 0.65, and 0.90. A pseudorandom number generator with a fixed seed was used to ensure reproducibility of the values.

It is essential to clarify that these observations do not originate from Mininet, NS-3, a hardware test bench, or equipment measurements. They are calculated from illustrative equations defined by the dataset's author. The random variations added around the central values represent artificial dispersion between replicates; they do not correspond to characterized experimental variability. Therefore, the data serve to illustrate a methodology, not to validate the system's performance.

The scenario's base consumption was defined as a function of network size and load: $B = 18 + 0.42N + 17L$, where N represents the number of nodes and L the load factor. For the conventional network, the core consumption is assumed to be B. For SDN alone, it is constructed from 0.91B, to which constant

UNDER PEER REVIEW IN IJAR

and load-dependent terms are added to represent the management cost. For the hybrid model, the core value uses 0.82B plus an additional term of 4.2 and a load term of 2L. These coefficients are construction assumptions, not parameters estimated from any equipment.

The other metrics were generated by analogous functions. Hypothetical convergence times are constructed to represent a shorter delay in the hybrid variant, while the CPU management load is assumed to be higher due to the replication and merging functions. Relative random variation is then applied to energy, latency, throughput, convergence time, overhead, and losses. This construction implies that the relationships observed in the results partly stem from the assumptions built into the generator.

### 2.4. Variables and indicators

The energy variable is consumption expressed in watt-hours over a conventional one-hour period (Wh/h). In a real-world study, it should be obtained through instrumental measurement or by integrating a validated power model. Simply multiplying an average power by one hour would only be appropriate if the power were actually measured or reliably estimated.

Average latency is expressed in milliseconds and throughput in megabits per second. Convergence time is expressed in milliseconds and, in synthetic gaming, represents an abstract indicator of stabilization delay. Control overhead is expressed in Mbit/s. CPU utilization represents the percentage of load on a management/control resource, and packet loss rate is expressed as a percentage. In a test bench, these measurements would require precise operational definitions: measurement points, sampling period, number of streams, packet size, clock synchronization method, and outlier handling.

The relative energy gain is calculated as follows: $G\ (\%) = 100 \times (E_{classic} - E_{candidate}) / E_{classic}$. A positive value indicates lower consumption than the conventional network in the compared scenario; a negative value would indicate higher consumption. This metric should not be interpreted independently of throughput and quality of service. A robust comparison should also present the energy per bit forwarded or the energy per request fulfilled, depending on the service being evaluated.

### 2.5. Data Analysis

The data were grouped by network size, load level, and architecture. For each group, the mean and standard deviation were calculated for energy. The means of the other indicators were used for descriptive comparison. In an experimental study, it would be preferable to calculate confidence intervals and conduct statistical tests tailored to the distribution of residuals and the replicate structure. However, hypothesis tests are not relevant for demonstrating a system effect from data whose values have been generated by equations that already account for some of the differences.

UNDER PEER REVIEW IN IJAR

The validation protocol to be implemented later should include several independent replications, a randomized execution order of the architectures, a warm-up period, failure scenarios, and controlled traffic loads. Configuration parameters must remain identical across architectures, except for variant-specific mechanisms. Software versions, topology, scripts, traffic profiles, and raw data should be archived to allow for the reproduction of results.

## 3. Results

### 3.1. Energy consumption at average load

Table 1 presents the average energy consumption figures at average load. For 20 nodes, the conventional network consumes 37,638 Wh, SDN alone 37,502 Wh, and the hybrid architecture 36,400 Wh. For 50 nodes, the values are 49,693, 49,252, and 46,801 Wh, respectively. For 100 nodes, they reach 70,981, 67,477, and 63,645 Wh.

Table 1. Synthetic energy consumption as a function of network size, at average load.

| Knots | Classic (Wh) | SDN alone (Wh) | SDN–CRDT (Wh) | Hybrid vs. classic gain |
|---|---|---|---|---|
| 20 | 37,638 | 37,502 | 36,400 | 3.3% |
| 50 | 49.693 | 49,252 | 46.801 | 5.8% |
| 100 | 70.981 | 67,477 | 63.645 | 10.3% |

The apparent gain of SDN alone compared to the traditional network is small at 20 and 50 nodes, then more pronounced at 100 nodes (see figure below). The hybrid model shows a larger gap in larger scenarios. This trend is consistent with the structure of the generation equations, which assign the hybrid model a proportional consumption component of 0.82B, compared to 1.00B for the traditional network. Therefore, it is not an independent empirical finding.

UNDER PEER REVIEW IN IJAR

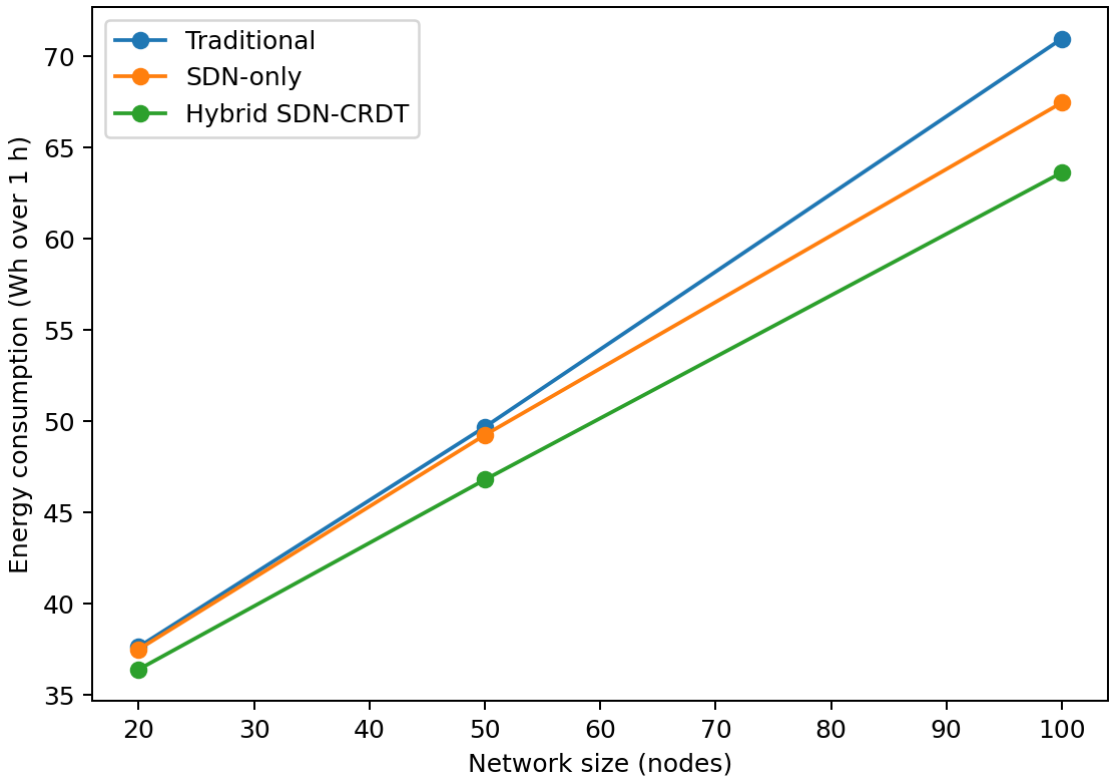



Figure 1. Synthetic energy consumption as a function of network size, at average load.

### 3.2. Energy Dispersion

The synthetic standard deviations of consumption at average load remain limited compared to the averages: for the conventional network, they are approximately 1.195 Wh, 1.719 Wh, and 1.947 Wh at 20, 50, and 100 nodes, respectively. For the hybrid model, they are 0.742 Wh, 0.798 Wh, and 1.124 Wh, respectively. These deviations reflect the level of random variation chosen in the generator. They cannot be interpreted as an estimate of the variability of a real network.

### 3.3. Latency and throughput

At medium load, for 100 nodes, the average synthetic latency reaches 25.633 ms in the traditional network, 21.397 ms in SDN alone, and 23.278 ms in the hybrid model. In this configuration, SDN alone exhibits the lowest latency, while the hybrid model falls between the other two architectures. The average synthetic throughput is close to 64.2 to 65.0 Mbit/s for all three variants. This proximity is partly due to the throughput formula, which uses the load level and a small correction related to the loss rate. It does not allow us to conclude that the performance is truly equivalent.

For the scenario with 100 nodes under high load, the synthetic latency is 28.307 ms for the traditional network, 23.286 ms for SDN alone, and 25.022 ms for the hybrid model. The average throughput remains close to 89 Mbit/s. These results illustrate the importance of simultaneously evaluating energy

UNDER PEER REVIEW

consumption and quality of service: one architecture can have lower power consumption while exhibiting higher latency than another architecture (see Figure 2).

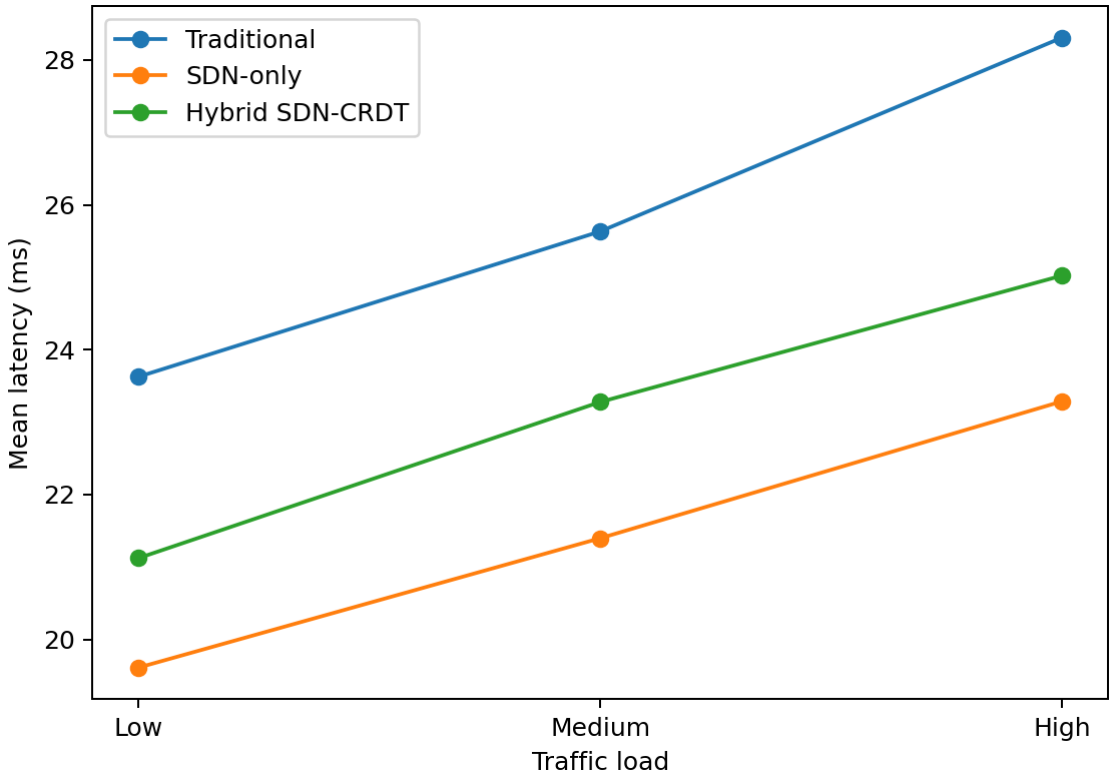



Figure 2. Synthetic mean latency as a function of traffic load, for 100 nodes.

### 3.4. Convergence time and control overhead

At average load and for 100 nodes, the synthetic convergence time is 434.001 ms for the conventional network, 318.056 ms for SDN alone, and 185.654 ms for the hybrid model. The corresponding average control overhead is 5.253, 4.138, and 5.148 Mbit/s, respectively. The hybrid model is thus associated, in the equations, with faster convergence than the other two variants, but its overhead is higher than that of SDN alone and close to that of the conventional network. This combination shows that a shorter convergence time does not necessarily imply lower control traffic.

### 3.5. CPU Load and Packet Loss

At medium load and for 100 nodes, synthetic CPU utilization is 56.03% for the traditional network, 67.81% for SDN alone, and 74.56% for the hybrid model. The higher load of the hybrid model is consistent with the assumption of additional cost associated with replication. However, the synthetic loss rate is lower for the hybrid model (0.565%) than for SDN alone (0.654%) and the traditional network (0.883%). These values are generator outputs and do not demonstrate that a CRDT reduces packet loss.

UNDER PEER REVIEW

Overall, the synthetic game produces a trend favoring hybrid architectures in terms of power consumption and convergence, at the cost of increased CPU load and higher overhead than SDN alone. This is an illustrative scenario designed to explore the trade-offs. Figure 2 above confirms the relevance of our tests. The ranking order could change if the energy coefficients, CRDT message cost, controller placement, or traffic profiles were modified.

## 4. Discussion

### 4.1. Cautious interpretation of apparent energy gain

The first observation (see Figure 1) is that the hybrid model exhibits lower overall power consumption than the conventional network across all three medium-load network sizes. The apparent gain increases from approximately 3.3% at 20 nodes to 5.8% at 50 nodes and 10.3% at 100 nodes. SDN alone shows an apparent gain of approximately 0.4%, 0.9%, and 4.9%, respectively. These differences are determined by the generation coefficients. They demonstrate how a model can represent an assumption of increasing benefit with size, but they do not prove that the CRDT mechanism actually generates these savings.

A physical interpretation would require breaking down the total energy into transfer energy, controller energy, management server consumption, replication traffic, and the cost of putting resources to sleep or waking them up. A hybrid architecture could save energy if the replicated control information sufficiently improves flow allocation, limits costly recalculations, or reduces time spent in inefficient states. Conversely, if updates are frequent or the replicated states are large, the synchronization cost can negate the benefits.

The literature on energy efficiency in SDN shows that consumption reduction often relies on concrete mechanisms for managing paths, rules, links, and equipment, and that it must be evaluated in conjunction with performance (Assefa & Özkasap, 2019). CRDT could facilitate the availability and convergence of information necessary for these mechanisms, but it does not replace either the optimization algorithm or the power management policy. The scientific contribution should therefore focus on how the replicated state improves or worsens a specific energy decision.

### 4.2. Scalability and replication cost

Increasing the number of nodes generally increases the volume of telemetry, the number of streams to manage, and the complexity of the state control. Delta-state CRDTs are advantageous when local changes can be represented by small increments, as they avoid systematically transmitting the entire state. Almeida et al. (2018) describe this framework and the conditions necessary to maintain

UNDER PEER REVIEW IN IJAR

convergence. However, the cost depends on the data structure, the update frequency, the number of replicas, and the propagation policy.

A scalability study should vary the number of switches, the number of controllers, the update frequency, and the size of the replicated state separately. It should compare full-state CRDTs with delta CRDTs, and then measure the bytes exchanged, CPU merge time, memory usage, convergence after partitioning, and energy consumption. A simple comparison by network size does not distinguish the effect of topology from that of the synchronization frequency.

### 4.3. Resilience and potential convergence

CRDTs can be useful in environments where replicas remain available during a temporary partition and merge their state when communication is restored, provided that the assumptions of CRDT and broadcasting are met (Shapiro et al., 2011). In a multi-controller SDN architecture, this property can support the continuity of monitoring data. However, it does not guarantee the availability of all paths, the absence of conflicting decisions, or the safety of a configuration applied by multiple controllers.

Resilience must be assessed using explicit scenarios: loss of a controller, partitioning of the control network, restoration of connectivity, message delays, duplication, and rebooting of a replica. Relevant metrics include detection time, traffic disruption duration, number of invalid rules, convergence time, and the amount of data exchanged during recovery. Security policies and routing invariants must be maintained regardless of data convergence.

### 4.4. Trade-off between energy, latency and CPU

The composite data illustrates a significant trade-off: the hybrid architecture consumes less power according to the model, but uses more CPU than the other variants. This result highlights that overall energy optimization cannot be derived from a single indicator. The additional CPU load may be acceptable if it avoids periods of energy-intensive operation on the transfer equipment, but it can also increase the power consumption of the control server. A credible measurement must integrate all system components, ideally with a clearly defined energy boundary.

Hybrid latency is also higher than that of SDN alone in synthetic scenarios. This penalty can arise, in a real-world implementation, from update propagation, metadata processing, or decision policy. Conversely, replication can reduce read latency by bringing the control state closer to the decision point. The sign of the trade-off therefore depends on the type of request, the controller topology, the broadcast protocol, and the semantics of the replicated data.

### 4.5. Limitations

UNDER PEER REVIEW IN IJAR

The main limitation is fundamental: the results presented are not from an executable network simulation, but from an equation generator. Means, standard deviations, and differences between architectures cannot be used as proof of efficiency. The power consumption equations impose some of the hybrid advantage from the outset; the convergence and CPU equations impose other trends. The dataset is useful for developing statistical processing and presentation, but it does not constitute a network benchmark.

A second limitation stems from the lack of a precise implementation. The CRDT type, replication protocol, topology, SDN controller, switching technology, routing rules, and energy measurement protocol are not defined by a single execution. A third limitation is the absence of experimental uncertainty analysis, appropriate statistical tests, and comparison with published baselines . Finally, comparing a traditional architecture to an SDN architecture must account for differences in functions and hardware; otherwise, the effects cannot be attributed to the tested mechanism.

### 4.6. Proposed Validation Protocol

Validation should first involve choosing an environment suited to the question. Mininet with Open vSwitch and a controller such as Ryu, ONOS, or OpenDaylight would allow for the study of software interactions and control messages; NS-3 might be preferred if the propagation model, queues, or characteristics of the simulated network are central. The choice must be justified, as a software emulator does not automatically provide a reliable measurement of power consumption.

The protocol must implement the three architectures under comparable conditions, define replicated status messages, and publish merge rules. Host and controller power consumption must be measured, or an explicit power model calibrated on hardware must be used. Scenarios should cover 20, 50, 100 nodes and larger, with varying loads, bursts, controller failures, and partitions. Each scenario should be repeated with independent seeds and a randomized execution order. Results should report mean, standard deviation, confidence interval, effect size, and trade-offs on latency and throughput.

An ablation experiment would be particularly informative: (i) SDN alone; (ii) SDN with classical replication; (iii) SDN with full-state CRDT; (iv) SDN with delta-CRDT; (v) hybrid with and without energy optimization policies. This comparison would allow us to distinguish the benefit of replication, the benefit of the CRDT type, and the benefit of the energy-saving algorithm. Without this decomposition, it would be difficult to attribute a gain to the SDN-CRDT integration itself.

### 4.7. Implications for IoT and 5G networks

IoT and edge environments often feature distributed nodes, intermittent links, and availability constraints. Low-coordination replication can be useful for sharing certain monitoring states or

UNDER PEER REVIEW IN IJAR

aggregated observations. In 5G networks and multi-access edge architectures, distributed orchestration can also benefit from a locally available control view. However, time-sensitive and safety-critical functions should not rely on eventual convergence without safeguards.

The most promising approach involves combining a CRDT (Control and Data Transfer Device) adapted to the data semantics, an explicit energy optimization policy, and a controller capable of verifying service constraints before applying an action. The potential benefits can then be tested: reduced synchronization time, improved control continuity, or shorter durations of inefficient states. These effects must be quantified on a representative infrastructure before any claims of superiority are made.

## 5. Conclusion

This article proposed an exploratory framework for analyzing the combination of SDN and CRDT in a network energy management strategy. SDN provides programmability and control abstraction, while CRDT can ensure the convergence of certain replicated data in the presence of concurrent updates. Their combination could facilitate the availability of the control state and reduce certain coordination costs, but CRDT does not directly constitute an energy-saving algorithm.

The synthetic dataset of 270 observations shows, by design, lower hybrid power consumption than a typical network at average load, with apparent reductions of 3.3%, 5.8%, and 10.3% for 20, 50, and 100 nodes, respectively. The model also indicates a shorter convergence time, but a higher CPU load and control overhead that is not consistently lower. These values illustrate possible trade-offs; they are not experimental results, as they are generated by hypothetical equations and not by a network simulator or test bench.

The current contribution is therefore methodological: defining the indicators, organizing the comparison, and formulating a validation protocol. The next step involves implementing the architecture, specifying the replicated data and the type of CRDT, measuring end-to-end energy, and comparing the results to reference architectures under reproducible conditions. Such validation, along with accessible scripts and data, would be necessary to support a scientific contribution submitted for Q2/Q3 review.

## Conflicts of interest

The authors declare no conflict of interest regarding the publication of this article.

## Bibliographical references (APA, 7th edition)

UNDER PEER REVIEW IN IJAR

Assefa, BG, and Özkasap , Ö. (2019). Studying energy efficiency in SDN networks: Software methods and optimization models. Journal of Network and Computer Applications, 137, 127–143. https://doi.org/10.1016/j.jnca.2019.04.001

Almeida, PS, Shoker, A., & Baquero, C. (2018). Replicated data types with delta state. Journal of Parallel and Distributed Computing, 111, 162–173. https://doi.org/10.1016/j.jpdc.2017.08.003

Haleplidis, E., Pentikousis , K., Denazis , S., Salim, J.H., Meyer, D., and Koufopavlou , O. (2015). Software-defined networking (SDN): Layers and architectural terminology (RFC 7426). Internet Research Working Group. https://www.rfc-editor.org/rfc/rfc7426

Kleppmann, M., and Beresford, A. R. (2017). A conflict-free replicated JSON data type. IEEE Transactions on Parallel and Distributed Systems, 28(10), 2733–2746. https://doi.org/10.1109/TPDS.2017.2697382

Kreutz, D., Ramos, FMV, Verissimo, PE, Rothenberg, CE, Azodolmolky , S., & Uhlig, S. (2015). Software-defined networks: A comprehensive study. Proceedings of the IEEE, 103(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). Conflict-free replicated data types. In SDL (Ed.), Stabilization, safety, and security of distributed systems (pp. 386–400). Springer. https://doi.org/10.1007/978-3-642-24550-3_29

**Data Transparency Note**

The associated data file has the status SYNTHETIC_ILLUSTRATIVE_NOT_MEASURED. It is suitable for testing scripts, practicing visualization, and preparing the structure of a paper. Before submission, the results in this version must be replaced with measurements from a reproducible network simulation or real-world experiments. Synthetic numerical values should not be cited as empirical scientific results.

SDN–CRDT Exploratory Manuscript | Synthetic Data —

UNDER PEER REVIEW IN IJAR