

In-Network Hamiltonian Engineering for 6G: Service Model, Quantitative Budgets, and Re-Entry Analysis

In-Network Hamiltonian Engineering for 6G: Service Model, Quantitative Budgets, and Re-Entry Analysis

Abstract

The QNFO program (In-Network Hamiltonian Engineering for 6G, <https://papers.qnfo.org/papers/in-network-hamiltonian-engineering-for-6g/>) proposes placing semi-classical Hamiltonian control logic inside the network fabric of sixth-generation (6G) telecommunications infrastructure, so that network nodes participate directly in quantum state and process control. This paper develops the concept in three quantitative steps. First, we define a Hamiltonian Engineering Service (HES): a contract between quantum endpoints and in-network controllers, executed at microsecond cadence against millisecond coherence budgets. Second, we derive the timing and fidelity budget of the service: for a node with coherence time $T_2 = 1 \text{ ms}$, an engineering cycle $T_c = 8 \mu\text{s}$, and a per-cycle error budget $\epsilon_c = 10^{-5}$, we compute $N_c = 125$ cycles per coherence window, accumulated error 1.25×10^{-3} , and a projected end-of-window fidelity ≈ 0.99875 under a stated linear accumulation model, at an actuation rate of $2 \times 10^6 \text{ s}^{-1}$ per endpoint. Third, we formalize re-entry — reuse of the control stack across infrastructure generations — as a measurable ratio, obtaining $\rho = 0.5$ for a stated twelve-component inventory, and a closed-loop latency of $1.8 \times 10^{-4} \text{ s}$ consuming $\eta = 0.18$ of the coherence window. Every number is computed here with shown arithmetic from explicitly labeled assumptions; none is an empirical measurement. We situate the framework in the Hamiltonian-engineering, quantum-network systems-engineering, and software-engineering literatures, and identify verification and benchmarking as the principal open problems.

1. Introduction

Sixth-generation network research increasingly treats the network not as a passive transport medium but as an active computational substrate. The QNFO concept pushes this trend to an extreme: it proposes to perform Hamiltonian engineering — the deliberate reshaping of an effective quantum Hamiltonian by a designed, time-dependent control Hamiltonian $H_c(t)$ applied on top of a natural Hamiltonian H_0 — inside the network itself, so that network nodes participate directly in quantum state and process control.

This creates a dependency inversion. Traditionally, quantum control hardware sits in the laboratory and the network merely carries results. In QNFO, the network is the control instrument. Two questions follow. First, is the service physically realizable at network timescales: can a control cycle executed through network elements close within a coherence window, and at what fidelity and actuation cost? Second, is the architecture maintainable: when the underlying 6G infrastructure is upgraded or replaced, can the control stack be re-entered — carried across — without full redesign?

Three observations motivate the placement of control in the network. First, quantum-secure telecommunications are already recognized as a systems-integration problem: realizing quantum-technology advances depends on engineering highly complex systems that integrate quantum devices into existing classical infrastructure [4]. Second, the control theory needed — using a semi-classical controller to engineer quantum Hamiltonians for state, process, or observable-optimization objectives — is generic and does not intrinsically require co-location [1]. Third, pulse-

sequence design has matured to where higher-order effects in the Floquet-Magnus expansion can be systematically accounted for with simple, intuitive decoupling rules [5], which is precisely the kind of compiled, rule-based artifact a network control plane can distribute and execute.

The paper makes three contributions: (i) a service model and quantitative timing/fidelity budget for the Hamiltonian Engineering Service; (ii) a closed-loop latency, bandwidth, and re-entry reuse analysis; (iii) a mapping onto the systems-engineering and software-engineering literatures with an explicit statement of what would falsify each claim. Every number is either computed here with shown arithmetic or explicitly labeled a projection with stated assumptions. We make no empirical claims about deployed 6G or quantum hardware.

2. Background and Related Work

We group the nine supplied works into three strands: quantum control theory, quantum-network systems engineering, and engineering-methodology analogies from software and web engineering.

Quantum control and Hamiltonian engineering. Reference [1] describes strategies for using a semi-classical controller to engineer quantum Hamiltonians in order to solve control problems such as quantum state or process engineering or optimization of observables. This is the theoretical foundation of the in-network approach: the controller is semi-classical, meaning it treats the quantum system as the object being steered while itself running on classical hardware, which is exactly the deployment position QNFO assigns to network elements. Reference [5] introduces a framework for designing Hamiltonian engineering pulse sequences that systematically accounts for higher-order contributions to the Floquet-Magnus expansion, yielding simple, intuitive decoupling rules despite the higher-order contributions naively involving complicated, non-local-in-time decoupling rules despite the higher-order contributions naively involving complicated, non-local-in-time commutators. A network control plane cannot host per-device numerical optimization at microsecond cadence; it needs precompiled, rule-based sequence templates whose error scaling is analytically understood, which is what [5] provides. Reference [2], from a different domain, analyzes the Jupiter–Saturn 2:5 near-commensurability in a fully analytic Hamiltonian planetary theory, with computations for the Sun–Jupiter–Saturn system extending to third order in the masses and eighth degree in the eccentricities and inclinations; these computations reveal an unexpectedly sensitive dependence of the solution on initial data and its likely nonconvergence. We cite it as a cautionary precedent from Hamiltonian methods: analytic perturbation treatments of near-resonant Hamiltonian systems can exhibit extreme sensitivity to initial conditions, and engineered effective Hamiltonians are likewise near-resonant constructions whose validity must be bounded, not assumed.

Quantum networks as engineered systems. Reference [4] explores evolving quantum key distribution (QKD) network architecture using a model-based systems-engineering approach, motivated by the growing need for quantum-secure telecommunications that overcome threats to encryption; its entry states that realization of quantum-technology advances in sensors, computing, timing, and communication depends on engineering highly complex systems that integrate quantum devices into existing classical infrastructure. The in-network approach extends this integration agenda one layer down: where [4] treats the QKD network architecture as the system under design, QNFO treats pulse-level control of the quantum endpoints as a service the network must host, schedule, and verify. The supplied summary of [4] is truncated and does not state which specific architecture-evolution results were obtained, so we cite it for its framing rather than for any specific finding.

Engineering-methodology analogies. Reference [6] defines Web Engineering as the application of systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of Web-based applications, framed as both a pro-active approach and a growing body of theoretical and empirical research. The analogy is deliberate: in-network quantum control needs the same discipline — quantifiable approaches to operation and

maintenance — that Web Engineering argued the Web needed, applied to control-plane software whose "runtime" is quantum state. Reference [8] addresses software engineering for collective cyber-physical ecosystems, characterized by dense and large networks of devices capable of computation, communication, and interaction with the environment and people; it notes that most research treats such systems as composites, i.e., heterogeneous functional complexes, while recent developments in fields such as self-organization (the entry's text is truncated at this point) are changing that picture. A network of quantum endpoints plus in-network controllers is precisely such a dense cyber-physical ecosystem, and the composite-versus-self-organizing tension maps onto the choice between centrally scheduled and locally autonomous pulse generation. Reference [3] surveys generative AI for software engineering, noting that the ability to automatically check, synthesize, and modify software engineering artifacts promises to revolutionize the field and that over a hundred LLM-based code models have been published since 2021; the entry's text is truncated mid-sentence and supplies no further findings. We invoke it only for the claim that automated artifact synthesis and checking is an active, large-scale capability that could in principle be pointed at pulse-sequence artifacts. Reference [7] argues that large language models for code are advancing fast while evaluation lags behind: current benchmarks focus on narrow tasks and single metrics, hiding critical gaps in robustness, interpretability, fairness, efficiency, and real-world usability, and suffer from inconsistent data engineering practices, limited software engineering context, and widespread contamination issues. This bears directly on the verification problem: if pulse-schedule generators become AI-assisted artifacts, the benchmarking pathologies catalogued in [7] would apply to software whose failure mode is silent decoherence rather than a wrong test result.

Finally, reference [9] is the QNFO program record itself (<https://papers.qnfo.org/papers/in-network-hamiltonian-engineering-for-6g/>); its supplied entry contains only the title and identifier, and this paper is the first substantive text produced under that identifier. All QNFO-specific technical content below is reconstructed from the control model of [1] plus the placement decision implied by the title, and every such reconstruction is stated as an assumption.

3. Methods

3.1 Service model

We define one new network service, the Hamiltonian Engineering Service (HES), with the following contract. A quantum endpoint E_q (a memory, transducer, or QKD node) exposes: (i) its natural Hamiltonian parameters $H_0(\theta)$, where θ is a vector of slowly drifting device parameters; (ii) a coherence time T_2 ; and (iii) a pulse-actuation interface accepting control waveforms with minimum pulse width τ_p . An in-network controller C_n (hosted on an edge node or smart network interface card) exposes: (i) a sequence-template library $\mathcal{L}$, each template L_k being a compiled decoupling or recoupling sequence in the sense of [5]; (ii) a scheduler that aligns cycle boundaries with network timing signals; and (iii) a verification module that checks the template's applicability conditions against the reported θ .

The service-level objective is stated as an effective-Hamiltonian error tolerance: over one engineering cycle of duration T_c , the stroboscopic dynamics must satisfy

$$\|\bar{H} - H_{\text{eff}}\| \leq \epsilon_c \|H_0\|,$$

where $\bar{H}$ is the cycle-averaged Hamiltonian and ϵ_c is the per-cycle relative error budget.

3.2 Timing and fidelity model

Each cycle consists of n_p pulses of width τ_p separated by idle intervals, so

$$T_c = n_p(\tau_p + \tau_i), \quad f = \frac{n_p \tau_p}{T_c} = \frac{\tau_p}{\tau_p + \tau_i},$$

with f the actuation duty factor. The controller must issue actuation updates at rate $R_u = 1/\tau_p$ per endpoint; a node hosting K endpoints presents an aggregate update rate $K R_u$.

We adopt a first-order accumulation model in which per-cycle fractional error contributions add linearly over N_c cycles within one coherence window:

$$N_c = \left\lfloor \frac{T_2}{T_c} \right\rfloor, \quad \epsilon_{\text{tot}} = N_c \epsilon_c, \quad F_{\text{proj}} \approx 1 - \epsilon_{\text{tot}},$$

where F_{proj} is the projected end-of-window process fidelity. This linear model is conservative relative to coherent-error growth models in the worst case and optimistic relative to models with cancellation; we flag it as a modeling assumption, not a measurement.

3.3 Closed-loop latency and re-entry models

We model one in-network feedback cycle as four sequential stages:

$$T_{\text{loop}} = T_s + T_p + T'_c + T_a,$$

where T_s is the sensing/measurement interval, T_p the packet transport latency from sensor to in-network compute node and back, T'_c the compute time for the control update, and T_a the actuation setup time. Re-entry is physically meaningful only if the loop closes within the coherence window:

$$\frac{T_{\text{loop}}}{T_2} \leq \eta_{\text{max}},$$

with η_{max} a stated design margin (we take $\eta_{\text{max}} = 0.2$).

For re-entry, let the QNFO stack consist of N_{tot} components partitioned into a physics-facing layer $\mathcal{P}$ (Hamiltonian targets, pulse sequences, control algorithms) and an infrastructure-facing layer $\mathcal{I}$ (timing distribution, transport, node hardware, qualification). The re-entry reuse ratio is

$$\rho = \frac{N_{\text{reuse}}}{N_{\text{tot}}}, \quad N_{\text{reuse}} = |\mathcal{P}_{\text{portable}}| + |\mathcal{I}_{\text{portable}}|,$$

where a component is portable if its specification does not reference generation-specific infrastructure parameters. The inventory is enumerated explicitly in Section 4 so that ρ is auditable.

4. Analysis

Every input number below is a design assumption of this paper, not an empirical measurement; each is labeled.

Inputs (timing/fidelity model). $T_2 = 1 \text{ ms} = 10^{-3} \text{ s}$ (A1); $\tau_p = 0.5 \mu\text{s} = 5 \times 10^{-7} \text{ s}$ (A2); $\tau_i = 0.5 \mu\text{s} = 5 \times 10^{-7} \text{ s}$ (A3); $n_p = 8$ (A4); $\epsilon_c = 10^{-5}$ (A5); $K = 64$ endpoints per node (A6).

Derivation 1 — cycle time.

$$T_c = n_p(\tau_p + \tau_i) = 8 \times (5 \times 10^{-7} + 5 \times 10^{-7}) \text{ s} = 8 \times 10^{-6} \text{ s} = 8 \mu\text{s}.$$

Derivation 2 — duty factor.

$$f = \frac{\tau_p}{\tau_p + \tau_i} = \frac{5 \times 10^{-7}}{10^{-6}} = 0.5.$$

Half of every coherence window is consumed by actuation; the endpoint is available for data operations only in the idle fraction $1 - f = 0.5$.

Derivation 3 — cycles per coherence window.

$$\frac{T_2}{T_c} = \frac{10^{-3}}{8 \times 10^{-6}} = 125, \quad N_c = \lfloor 125 \rfloor = 125.$$

Derivation 4 — accumulated error and projected fidelity.

$$\epsilon_{\text{tot}} = N_c \epsilon_c = 125 \times 10^{-5} = 1.25 \times 10^{-3}, \quad F_{\text{proj}} \approx 1 - 1.25 \times 10^{-3} = 0.99875.$$

Derivation 5 — actuation update rate per endpoint.

$$R_u = \frac{1}{\tau_p} = \frac{1}{5 \times 10^{-7} \text{ s}} = 2 \times 10^6 \text{ s}^{-1}.$$

Derivation 6 — aggregate update rate per node.

$$K R_u = 64 \times 2 \times 10^6 \text{ s}^{-1} = 1.28 \times 10^8 \text{ s}^{-1}.$$

Derivation 7 — sensitivity to T_2 . If T_2 degrades to $T'_2 = 10^{-4} \text{ s}$, then

$$N'_c = \left\lfloor \frac{10^{-4}}{8 \times 10^{-6}} \right\rfloor = 12, \quad \epsilon'_{\text{tot}} = 12 \times 10^{-5} = 1.2 \times 10^{-4}, \quad F'_{\text{proj}} \approx 0.99988.$$

Note the counterintuitive direction: shorter T_2 reduces accumulated error because fewer cycles execute before the window closes, but it also reduces the useful window by the same factor — the fidelity metric alone understates the cost of decoherence, so we report N_c alongside F_{proj} .

Derivation 8 — error budget headroom. To hold $\epsilon_{\text{tot}} \leq 0.01$ (adopted here as a design target, not a community standard) with $N_c = 125$:

$$\epsilon_c \leq \frac{0.01}{125} = 8 \times 10^{-5},$$

so the chosen $\epsilon_c = 10^{-5}$ carries a headroom factor of $8 \times 10^{-5} / 10^{-5} = 8$.

Inputs (closed-loop and re-entry models). $T_s = 1.0 \times 10^{-4} \text{ s}$ (A7); $T_p = 2.0 \times 10^{-5} \text{ s}$ (A8); $T'_c = 5.0 \times 10^{-5} \text{ s}$ (A9); $T_a = 1.0 \times 10^{-5} \text{ s}$ (A10); $W = 64$ bits per qubit per control word (A11); $N_q = 1000$ qubits (A12); $R_{\text{link}} = 1.0 \times 10^{11} \text{ bit/s}$ (A13); $\eta_{\text{max}} = 0.2$ (A14).

Derivation 9 — loop latency.

$$T_{\text{loop}} = 1.0 \times 10^{-4} + 2.0 \times 10^{-5} + 5.0 \times 10^{-5} + 1.0 \times 10^{-5} = 1.8 \times 10^{-4} \text{ s}.$$

Derivation 10 — coherence fraction.

$$\eta = \frac{T_{\text{loop}}}{T_2} = \frac{1.8 \times 10^{-4}}{1.0 \times 10^{-3}} = 0.18 \leq \eta_{\text{max}} = 0.2,$$

with margin $\eta_{\max} - \eta = 0.02$ and headroom factor $\eta_{\max}/\eta = 0.2/0.18 \approx 1.111$. The margin is thin: an increase of about 11% in any stage latency exhausts it.

Derivation 11 — control-plane bandwidth per cycle.

$$B_{\text{cycle}} = W \cdot N_q = 64 \times 1000 = 6.4 \times 10^4 \text{ bits}, \quad \tau_{\text{tx}} = \frac{6.4 \times 10^4}{1.0 \times 10^{11}} = 6.4 \times 10^{-7} \text{ s}.$$

The transmission time is negligible relative to T_{loop} : $\tau_{\text{tx}}/T_{\text{loop}} = 6.4 \times 10^{-7}/1.8 \times 10^{-4} \approx 3.56 \times 10^{-3}$, i.e., about 0.36% of the loop budget. Under these assumptions, bandwidth is not the binding constraint; latency staging is.

Derivation 12 — re-entry reuse ratio. We enumerate a twelve-component inventory of the QNFO stack (Assumption A15; the partition is ours, since [9]'s supplied context gives no component list):

Portable count: components 1–5 and 12 give $N_{\text{reuse}} = 6$; $N_{\text{tot}} = 12$; therefore

$$\rho = \frac{6}{12} = 0.5.$$

Counting component 11 as portable would give $\rho = 7/12 \approx 0.583$; we report the conservative $\rho = 0.5$ as the headline and $0.5 \leq \rho \leq 0.583$ as the bound.

Derivation 13 — re-qualification effort (projection). If re-qualification effort per non-portable component is $e = 4$ person-weeks (Assumption A16), then

$$E_{\text{req}} = e (N_{\text{tot}} - N_{\text{reuse}}) = 4 \times 6 = 24 \text{ person-weeks}.$$

This is a projection under A16; its uncertainty is linear in e , giving $E_{\text{req}} \in [12, 48]$ person-weeks for $e \in [2, 8]$.

Divergent projection — aggregate control bandwidth (labeled projection, not adopted as a headline). An alternative aggregate model assumes $N = 1000$ nodes, each repeating a 256-bit encoded pulse sequence at $f = 1$ MHz, giving

$$B_{\text{total}} = N \times f \times b = 1000 \times 10^6 \times 256 = 2.56 \times 10^{11} \text{ bit/s} = 256 \text{ Gb/s},$$

with independent 20% uncertainties on N , f , and b combined in quadrature:

$$\epsilon = \sqrt{3 \times 0.20^2} = \sqrt{0.12} \approx 0.346, \quad B_{\text{total}} \in [256 \times 0.654, 256 \times 1.346] \approx [167, 345] \text{ Gb/s}.$$

This projection rests on assumptions (simultaneous per-node repetition at 1 MHz across 1000 nodes) that are stronger than those of Derivations 5–6; we report it in Appendix A as a documented divergence rather than as a result, together with the claim-attribution table in Appendix B.

5. Results

All results below follow from the derivations in Section 4 under the stated assumptions A1–A16; none is an empirical measurement. Where "projected" appears, the projection is the linear accumulation model of Section 3.2 applied to the stated inputs.

Quantity	Symbol	Value
Engineering cycle time	T_c	$8 \times 10^{-6} \text{ s}$

Quantity	Symbol	Value
Actuation duty factor	f	0.5
Cycles per coherence window	N_c	125
Accumulated error over one window	ϵ_{tot}	1.25×10^{-3}
Projected end-of-window fidelity	F_{proj}	≈ 0.99875 (projection; linear model)
Update rate per endpoint	R_u	$2 \times 10^6 \text{ s}^{-1}$
Aggregate update rate, $K = 64$	$K R_u$	$1.28 \times 10^8 \text{ s}^{-1}$
Per-cycle budget for $\epsilon_{\text{tot}} \leq 0.01$	ϵ_c^{max}	8×10^{-5}
Headroom factor at $\epsilon_c = 10^{-5}$	—	8
Degraded- T_2 case ($T'_2 = 10^{-4} \text{ s}$)	N'_c, F'_{proj}	12, ≈ 0.99988 (projection)
Closed-loop latency	T_{loop}	$1.8 \times 10^{-4} \text{ s}$
Coherence fraction consumed	η	0.18 (headroom ≈ 1.111)
Per-cycle control payload	B_{cycle}	6.4×10^4 bits ($\approx 0.36\%$ of loop budget)
Re-entry reuse ratio	ρ	0.5 (bound 0.5–0.583)
Projected re-qualification effort	E_{req}	24 person-weeks (range 12–48, projection)

The headline architectural conclusions are: (i) a microsecond-scale engineering cycle is compatible with millisecond-scale coherence under a 10^{-5} per-cycle error budget, but the required actuation rate of $2 \times 10^6 \text{ s}^{-1}$ per endpoint places the control function in the category of hardware-adjacent network compute, not application-layer software; (ii) under the closed-loop assumptions, latency staging rather than bandwidth is the binding constraint; (iii) re-entry is a half-rebuild: the physics-facing layer is portable, the infrastructure-facing layer — above all the timing plane — is not.

6. Discussion

Limitations of the fidelity model. The linear error-accumulation model (Section 3.2) is the weakest link. Coherent errors from a repeated non-echoed average-Hamiltonian term can add quadratically or with constructive phase, making ϵ_{tot} larger than $N_c \epsilon_c$; conversely, echo-structured sequences can cancel first-order terms, making it smaller. Reference [5] provides exactly the framework — systematic accounting of higher-order Floquet-Magnus contributions — that would replace the linear model with a commutator-based bound, and adopting it is the first remediation step. Until then, $F_{\text{proj}} = 0.99875$ should be read as a budget-allocation statement, not a fidelity prediction.

Limitations of the parameterization. Every input number is a design assumption. Real endpoints vary by orders of magnitude in T_2 , and Derivation 7 shows the results move accordingly. The claim that survives parameter variation is structural: $N_c = T_2/T_c$ and $\epsilon_{\text{tot}} = N_c \epsilon_c$ hold under the model for any inputs, so the architectural conclusion (microsecond cycles, hardware-adjacent control) follows whenever T_2 is in the millisecond range and ϵ_c is at or below the 10^{-4} level. The closed-loop model ignores serialization, queueing under load, and error-correction overhead in the control plane; any of these could dominate T_p and T'_c . The component inventory behind $\rho = 0.5$ is our reconstruction, not a documented QNFO design, since [9]’s supplied context gives only a title and DOI.

Failure modes. Three are foreseen. (i) Synchronization failure: if network timing jitter exceeds a fraction of $T_c = 8 \mu\text{s}$, cycle boundaries blur and the stroboscopic averaging argument collapses. (ii) Template mismatch: a compiled sequence L_k applied outside its validity domain produces an engineered Hamiltonian that is confidently

wrong — the near-resonant sensitivity precedent of [2] warns that analytic Hamiltonian constructions can fail nonconvergently rather than gracefully; we stress that [2] concerns planetary dynamics, not control loops, and the transfer is an analogy. (iii) Verification debt: if sequence templates are synthesized by generative tools in the manner of [3], the evaluation pathologies of [7] — narrow metrics, contamination, hidden robustness gaps — transfer to a setting where the test oracle is quantum state itself, which is far harder to observe than a unit-test suite.

What would falsify the claims. The timing/fidelity results are falsified by measured stage latencies, coherence times, or per-cycle errors differing materially from A1–A5. The realizability claim would be falsified by demonstrating that network-grade timing distribution cannot deliver jitter well below T_c , or that per-cycle relative errors at the 10^{-5} level cannot be achieved by any compiled rule-based sequence family on realistic H_0 spectra. The re-entry claim ($\rho = 0.5$) is falsified by a documented component inventory whose portable fraction differs from $6/12$, or by a migration in which the timing and transport planes carried over unchanged. The architectural claim that in-network placement yields operability benefits is falsified if co-located control proves strictly simpler and cheaper at $K = 64$ scale. The transfer of [2]’s sensitivity lesson to control loops could be falsified by showing that closed-loop feedback suppresses exactly the initial-condition sensitivity that open-loop analytic propagation exhibits.

Open questions. (i) What is the minimal interface contract between the physics-facing and infrastructure-facing layers that would raise ρ toward 1? (ii) How should the sequence-template library $\mathcal{L}$ be versioned, certified, and rolled back across a deployed network — the operation-and-maintenance discipline that [6] argued Web applications required? (iii) Can re-entry qualification be expressed as a benchmark that avoids the pathologies catalogued in [7], possibly using AI-assisted artifact checking in the sense of [3] (an explicitly unverified hypothesis; the supplied summary of [3] does not evaluate this use case)? (iv) Can the self-organizing alternatives to composite architectures noted in [8] be applied to pulse scheduling, replacing central schedulers with locally negotiated cycle boundaries?

7. Conclusion

We reframed in-network Hamiltonian engineering as a measurable service and re-entry problem. Under explicit assumptions, a $T_c = 8 \mu s$ cycle with a 10^{-5} per-cycle error budget admits $N_c = 125$ cycles per $T_2 = 1 ms$ coherence window with projected accumulated error 1.25×10^{-3} , at an actuation cost of 2×10^6 updates per second per endpoint; the closed-loop latency budget closes with a thin margin ($\eta = 0.18$ against $\eta_{\max} = 0.2$); and roughly half of the control stack ($\rho = 0.5$, bounded by 0.583) is portable across infrastructure generations, with a projected re-qualification effort of 24 person-weeks (range 12–48). The framework’s value lies less in these particular numbers — design assumptions run through deliberately simple models — than in the budget structure they instantiate: cycle time, per-cycle error, coherence window, and actuation rate are the quantities a 6G architecture must co-manage if quantum devices become infrastructure. The immediate agenda is to replace the linear accumulation model with Floquet-Magnus bounds of the kind developed in [5], to ground the systems argument in the model-based QKD-network methodology of [4], and to build the verification and benchmarking discipline that the software-engineering literature of [3], [6], [7], and [8] shows is always the lagging half of any new engineering capability.

References

-
- [1] Hamiltonian engineering for quantum systems. arXiv:quant-ph/0602014v2. <https://arxiv.org/abs/quant-ph/0602014v2> [2] The Great Inequality In A Hamiltonian Planetary Theory. arXiv:chao-dyn/9311011v2. <https://arxiv.org/abs/chao-dyn/9311011v2> [3] Morescent GAI for Software Engineering (Extended Version). arXiv:2406.04710v2. <https://arxiv.org/abs/2406.04710v2> [4] Exploration of Evolving Quantum Key Distribution

Network Architecture Using Model-Based Systems Engineering. arXiv:2508.15733v1. <https://arxiv.org/abs/2508.15733v1> [5] Higher-Order Methods for Hamiltonian Engineering Pulse Sequence Design. arXiv:2303.07374v1. <https://arxiv.org/abs/2303.07374v1> [6] Web Engineering. arXiv:cs/0306108v1. <https://arxiv.org/abs/cs/0306108v1> [7] Towards Comprehensive Benchmarking Infrastructure for LLMs In Software Engineering. arXiv:2601.21070v1. <https://arxiv.org/abs/2601.21070v1> [8] Software Engineering for Collective Cyber-Physical Ecosystems. arXiv:2406.04780v1. <https://arxiv.org/abs/2406.04780v1> [9] <https://papers.qnfo.org/papers/in-network-hamiltonian-engineering-for-6g/>. QNFO: In-Network Hamiltonian Engineering for 6G.

Appendix A. Divergence report

The aggregate control-bandwidth projection of Section 4 (Derivation 12, labeled projection) diverges from the per-endpoint actuation model of Derivations 5–6, and both conventions are documented here rather than silently merged.

- Convention A (adopted for headline results, Derivations 5–6): actuation demand is expressed per endpoint as $R_u = 1/\tau_p = 2 \times 10^6 \text{ s}^{-1}$ and per node as $K R_u = 1.28 \times 10^8 \text{ s}^{-1}$ for $K = 64$ endpoints (A6). This convention ties bandwidth to the pulse width τ_p and scales with deployed endpoints.
- Convention B (divergent projection): actuation demand is expressed as an aggregate network load $B_{\text{total}} = N \times f \times b = 1000 \times 10^6 \times 256 = 2.56 \times 10^{11} \text{ bit/s} = 256 \text{ Gb/s}$, with independent 20% uncertainties on N , f , and b combined in quadrature, $\epsilon = \sqrt{3 \times 0.20^2} = \sqrt{0.12} \approx 0.346$, giving $B_{\text{total}} \in [167, 345] \text{ Gb/s}$. This convention assumes simultaneous per-node repetition of a 256-bit sequence at $f = 1 \text{ MHz}$ across $N = 1000$ nodes, a stronger assumption than Convention A's.

The disagreement is one of accounting convention, not arithmetic: Convention A measures control-word updates at the endpoint interface, Convention B measures encoded-sequence traffic aggregated over the network. Convention A is adopted in the main text because its inputs (A2, A6) are shared with the timing model; Convention B is retained here as a projection because network-level aggregation may matter for transport planning. The two are not directly comparable without a mapping from τ_p -driven updates to encoded-sequence repetitions, which this paper does not supply.

Appendix B. Claim attribution

ID	Claim	Source drafts	Status
C1	HES service contract: endpoint exposes $H_0(\theta)$, T_2 , pulse interface; controller exposes template library, scheduler, verification	A, B	CONVERGENT
C2	Timing/fidelity model: $T_c = 8 \mu\text{s}$, $N_c = 125$, $\epsilon_{\text{tot}} = 1.25 \times 10^{-3}$, $F_{\text{proj}} \approx 0.99875$ under linear accumulation	A, B, C	CONVERGENT
C3	Actuation rates $R_u = 2 \times 10^6 \text{ s}^{-1}$ per endpoint, $K R_u = 1.28 \times 10^8 \text{ s}^{-1}$ per node ($K = 64$)	A, B	CONVERGENT
C4	Closed-loop latency $T_{\text{loop}} = 1.8 \times 10^{-4} \text{ s}$, $\eta = 0.18 \leq 0.2$	A, B, C	CONVERGENT
C5	Bandwidth non-binding: $\tau_{\text{tx}}/T_{\text{loop}} \approx 3.56 \times 10^{-3}$	A, B	CONVERGENT
C6	Re-entry ratio $\rho = 0.5$ from twelve-component inventory, bound 0.5–0.583	A, B, C	CONVERGENT
C7	Re-qualification effort projection $E_{\text{req}} = 24$ person-weeks (range 12–48)	B, C	CONVERGENT

ID	Claim	Source drafts	Status
C8	Aggregate bandwidth projection 256 Gb/s , range $[167, 345]$ Gb/s	C only	SINGLE (retained as labeled projection in Appendix A, not a headline)
C9	Degraded- T_2 sensitivity case ($N'_c = 12$, $F'_{\text{proj}} \approx 0.99988$) with counterintuitive direction noted	A, B	CONVERGENT
C10	Error-budget headroom factor 8 at $\epsilon_c = 10^{-5}$ for $\epsilon_{\text{tot}} \leq 0.01$	B only	SINGLE
C11	Falsification conditions and failure modes (synchronization, template mismatch, verification debt)	A, B, C	CONVERGENT
C12	Analogy to [2]'s near-resonant sensitivity, explicitly flagged as a cross-domain analogy	A, B	CONVERGENT