Galicia Supercomputing Center (CESGA), Santiago de Compostela, Spain

Distributed Quantum Computing at CESGA

Iago Fernández Llovo

Postdoctoral Research Fellow @ CESGA
Guest Researcher @ University of Innsbruck







AQADOC Workshop, Palaiseau 15 October 2026













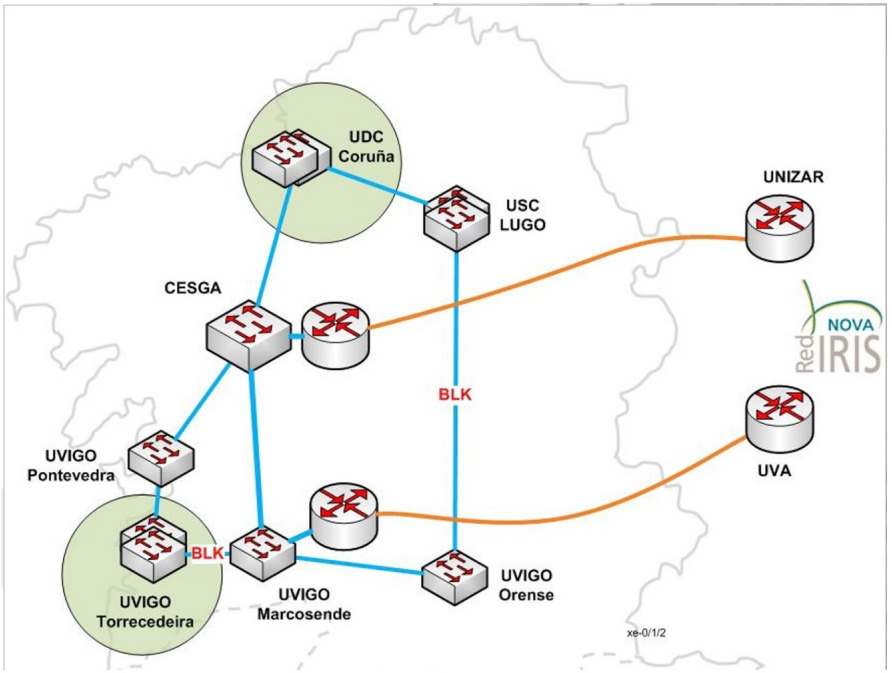


Galicia Supercomputing Center (CESGA)

1. Public research foundation, **1993**. Xunta de Galicia + CSIC. Part of **RES**, the Spanish Supercomputing Network

2. Operates **RECETGA** → Galician science and technology network

3. Four strategic axes:
   - Computer science
   - Earth sciences
   - Life sciences
   - Quantum technologies



Source: CESGA Anuario 2025

A quantum datacentre





22,848
cores + 157 GPUs, FinisTerrae III

224
user accounts in 2025, 17 projects via RES

32-qubit QPU
Qmio, open since 2024

164,194
jobs on Qmio in 2025
1,334 QPU hours

QEX
Applications, benchmarking and training on European quantum platforms
`EuroHPC JU`

DEEP QUANTUM
Protocols, partitioning and emulation for distributed quantum computing
`National grant`

1HealthAI
Europe's AI factory for global health. Includes **FinisTerrae IV and two quantum systems**
`82 M€ awarded in 2025`

Other projects: QATALYZE, GRAVITEQA, EQUIVALEMT, QUANTUM_IBER_IA, IberianQCI ...

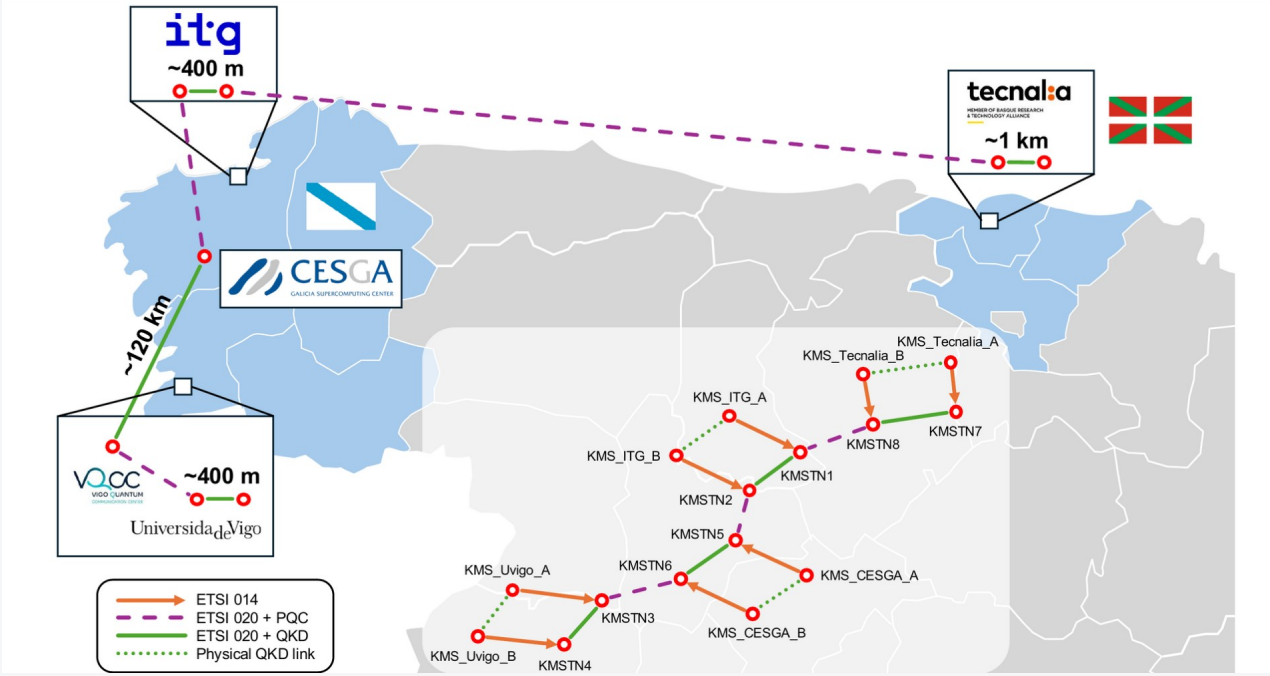
Source: CESGA Anuario 2025
IQM / Telefónica announcement, December 2025

Quantum networks



Santiago → Vigo: 120 km. Longest terrestrial quantum link in Spain (Oct 2025) with VQCC, UVigo. Fibre from Red.es

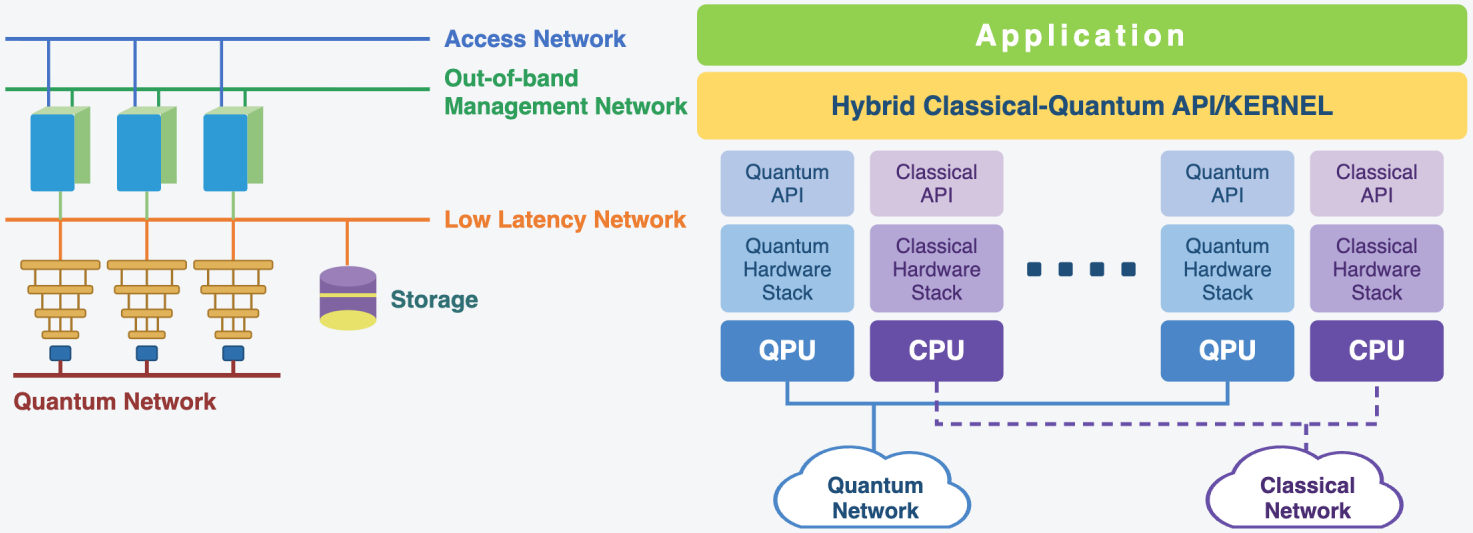
QUANTUM_IBER_IA. Interreg POCTEP, coordinated by GAIN, **17 partners in Spain and Portugal**: supercomputers, quantum networks, AI platforms. Link continues as **IberianQCI** → Portugal, inside EuroQCI



Key management across isolated QKD domains. Eight nodes, four sites, two regions ~500 km apart. Applications ask their local node for keys, never touch QKD hardware. Keys relayed hop by hop over ETSI GS QKD 014/020 → QKD where a quantum link exists, Kyber post-quantum where only classical WAN does

D. Barral, et al., arXiv:2604.20376 (2026)
KMSTN code: doi.org/10.5281/zenodo.20713366

Integrating a QPU into a datacentre: the vision



QPU as accelerator, not service

Offloaded kernels from FinisTerrae III → one access network, in-house middleware

Single queue & filesystem

Hybrid jobs scheduled as classical + quantum resources. No round trip outside the datacentre

Latency budget

Shot < 500 µs. Submission through SLURM, 1–3 s. → hold the QPU, stream circuits over a message bus

Open for DQC

Several QPUs → how are they co-scheduled? One network, many users → can it be partitioned?

CESGA contribution to DQC

1 **Network design**
Routing devices can assist on the computation → fewer ebits, shallower circuits

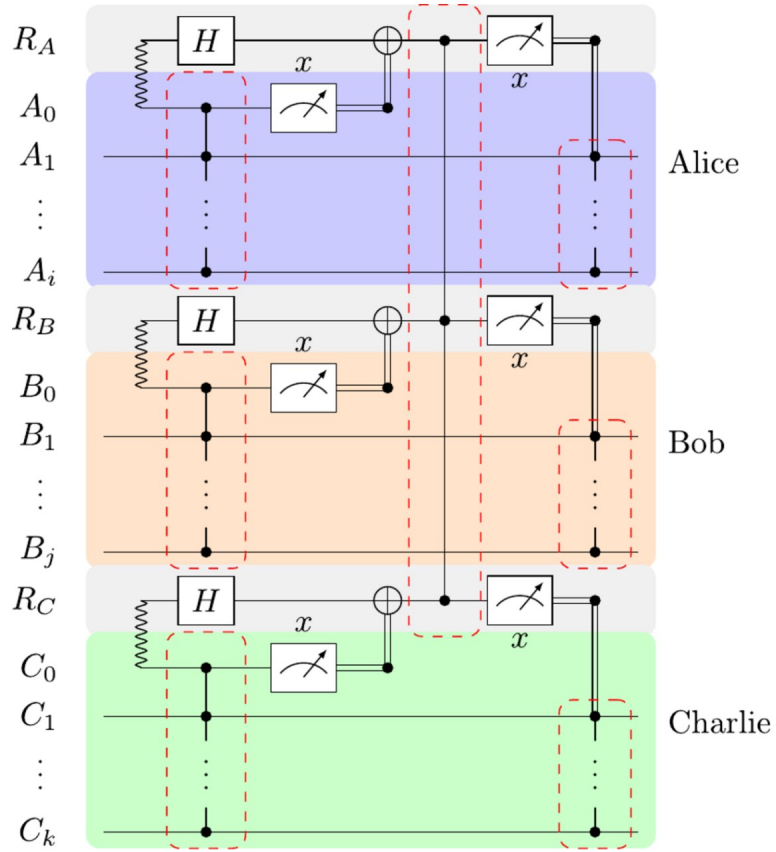
2 **Runtime and programming model**
vQPUs raised and scheduled like any HPC resource in the queue system (CUNQA)
User writes the algorithm → compiler and runtime pick protocol and route (NetQIR, NetQMPI)

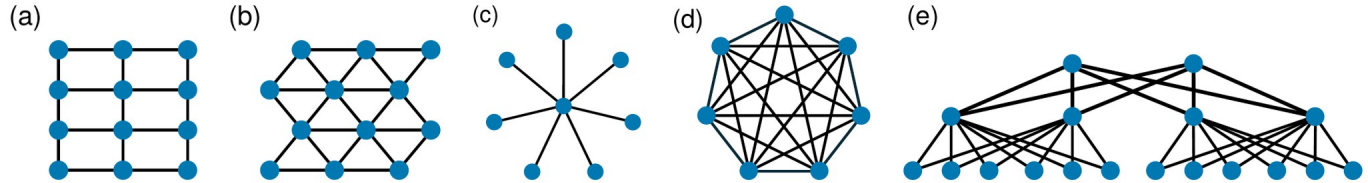
3 **Procurement toolbox**
Standard benchmarks for emulators now and for future machines → informed procurement

4 **A quantum network node**
Developing hardware that can be entangled on demand. We are ensuring this happens!

1. Quantum networks as design choices

Datacentre topology is known beforehand. We asked ourselves: **what tasks should routing devices perform?**





Beyond entanglement swapping

DQC routers can do more! We must explore what possibilities quantum networking brings to the table

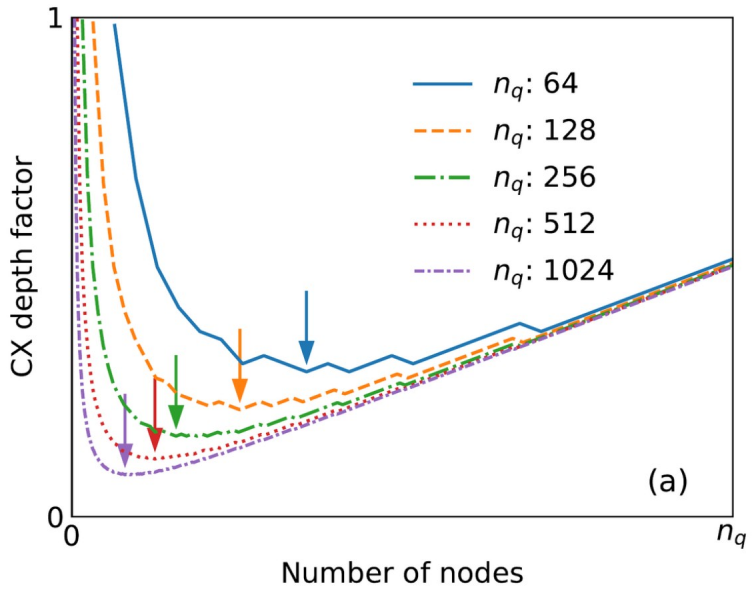
Network-assisted computation

One link per node (star network). Preshared entanglement, LOCC

Single point of failure, bottleneck as k grows → multi-tier routers are future work

Collective operations instead of point-to-point

One fan-out carries all **commuting diagonal gates** → a whole QAOA cost layer in one round
Grover: split an N-qubit MCZ over k nodes → CX depth **O(12N/k + 6k)** → **an optimum in k**



Depth against monolithic. Arrows: optimum

CROSS-QPU OPERATIONS

$O(k^2M^2) \rightarrow kM$

for k QPUs, M communication qubits each

NONLOCAL ROUNDS

several → one

swap-and-teleport opens a new round once cross-QPU gates outnumber communication qubits

OPTIMAL SPLIT

$k = \sqrt{(2N)}$

speedup $O(\sqrt{N})$, **12× at 1024 qubits**
2k ebits per Grover layer

CONNECTING SMALLER NODES

Smaller devices are easier to build and operate
Nodes join or leave by connecting to the router

QAOA landscapes reproduce the monolithic run
Partitioning is star-like → in unstructured graph savings may be lower (or none)

2. CUNQA: vQPUs allocated as an HPC resource

vQPU = classical resources + simulation software, raised as a job on the cluster. Integrated in queue system as QPUs

```
$ qraise -n 4 -t 00:10:00 --quantum_comm
$ qinfo
$ qdrop --all
```

(First) open source tool to emulate all three DQC communication models in one place



Server

Listens for incoming tasks

Backend

Qubit count, coupling graph, instruction set, noise models...
Same description as physical QPU
Circuit transpilation

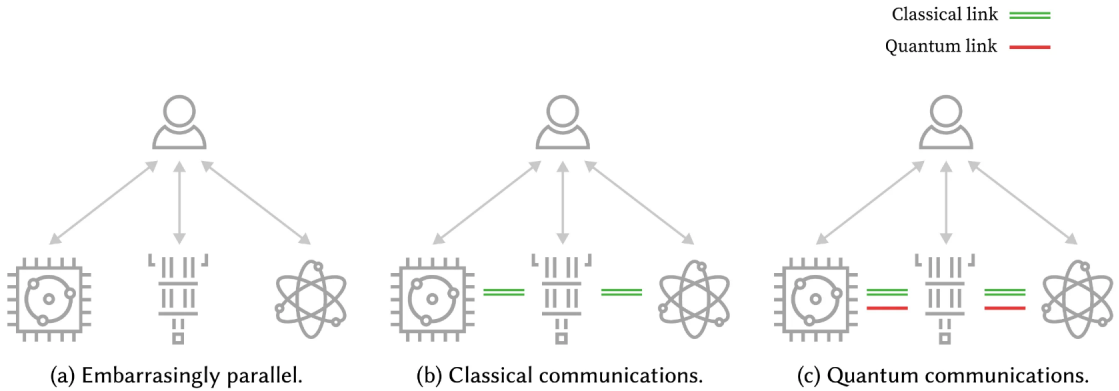
Classical simulators

- **Qiskit Aer**, default; GPU and noise
- MQT-DDSIM (Munich Quantum Toolbox)
- **CunqaSimulator**
- Qulacs
- ... and others

J. Vázquez-Pérez, et al., arXiv:2511.05209 (2025)
github.com/CESGA-Quantum-Spain/cunqa

Communication models in CUNQA

DQC MODEL	EMULATION STRATEGY	NOISE
Embarrasingly parallel	Independent vQPUs; distributed shots	Yes, per-qubit
Classical communications	Bits shared at runtime over ZMQ or MPI → gates conditioned on another vQPU's measurement	Work in progress (soon!)
Quantum communications	Entangled links cannot be split → one executor simulates the combined circuit, returns aggregated counts	Work in progress (soon!)



(a) Embarrasingly parallel. (b) Classical communications. (c) Quantum communications.

LIMITS

30 noisy qubits (distributing shots) with state vector, lower with DM
Entanglement costly! → QPE 13,975 s vs 291 s on one vQPU

DEPLOYED

CESGA, ComputaEx, BSC, CiTIUS, Julich

J. Vázquez-Pérez, et al., arXiv:2511.05209 (2025)
github.com/CESGA-Quantum-Spain/cunqa

NetQIR and NetQMPI

Users should not allocate entanglement by hand. **NetQIR** adds communication directives to QIR; **NetQMPI** as frontend

```
$ netqmpi -np 2 p2p.py
rank = comm.rank
if rank == 0:
    circuit.h(0)
    comm.qsend(circuit, [0], 1)
elif rank == 1:
    comm.qrecv(circuit, [0], 0)
```

An MPI model for DQC

One script across N nodes. No entanglement or teleportation allocated by hand

No-cloning removes broadcast

Expose shares a reference through entanglement. Scatter, gather, reduce survive

Compiler picks the protocol

Telegate vs teledata recorded in the IR → a cost model decides, not the user

Backend-agnostic

Same application on SquidASM, **CUNQA**, Aer or Qoala, chosen by a flag

Centro Singular de Investigación en Tecnoloxías Intelixentes

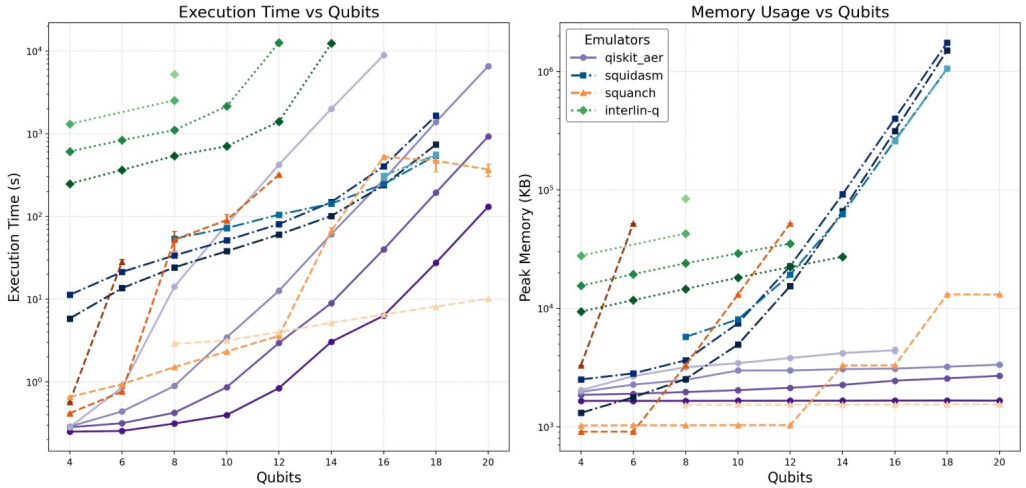
USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



F. J. Cardama, et al., *Future Gener. Comput. Syst.* **174** 107989 (2026)
F. J. Cardama, et al., *IEEE Access* **14** 125459 (2026)

3. How to compare DQC emulators (and systems!)?

DQC claims are validated in emulators. Most either lack teleportation primitives or need workarounds
We **buy quantum systems** → similar framework needed to judge the machines!



4 to 20 qubits; shade = 1, 2, 4 or 8 nodes. Three thetas, 100 shots

TEST CASE

Distributed inverse QFT, partitioned with gate teleportation

MEASURED

Execution time, memory, fidelity against a monolithic baseline

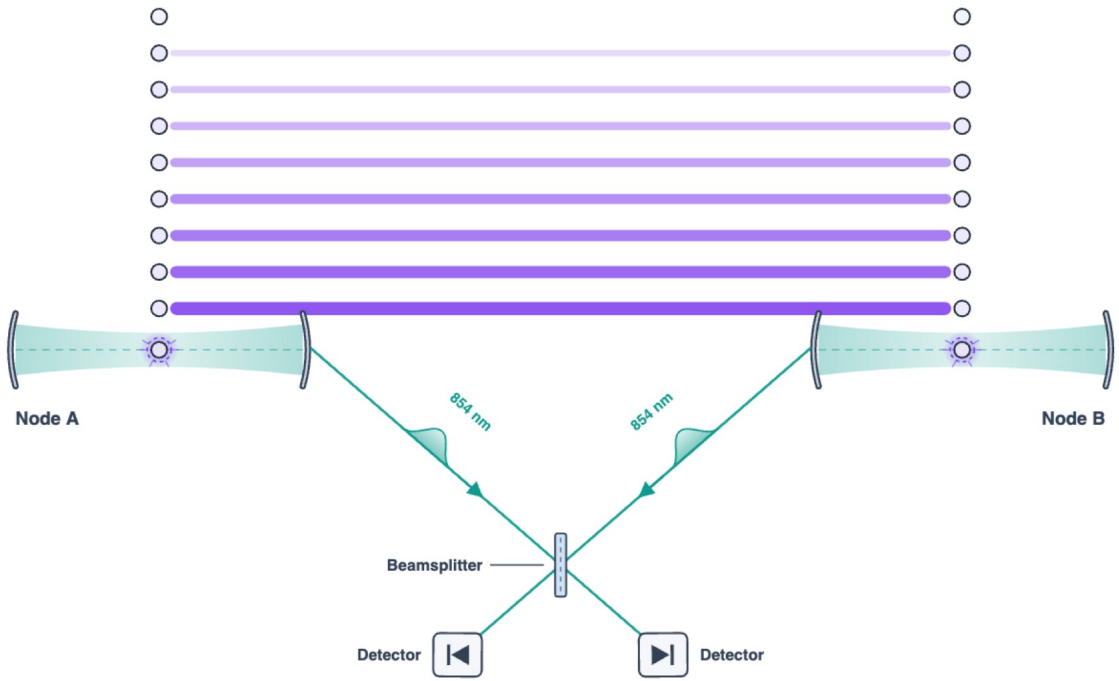
FOUR EMULATORS

Qiskit Aer, SquidASM, Interlin-q & **SQUANCH**

RESULT

Fidelity vs scalability. SquidASM models the network properly, with some overhead, but scales better
Lighter tools seem fast but lose fidelity or crash. → We need faithful DQC emulation at HPC scale

4. Growing entanglement between quantum network nodes



Each node: **$^{40}Ca^{+}$ in a cavity-integrated trap** → cavity-enhanced Raman process → 854 nm photon. The two photons interfere on a beamsplitter; a coincident detection **heralds entanglement between the ions**

OBJECTIVES:

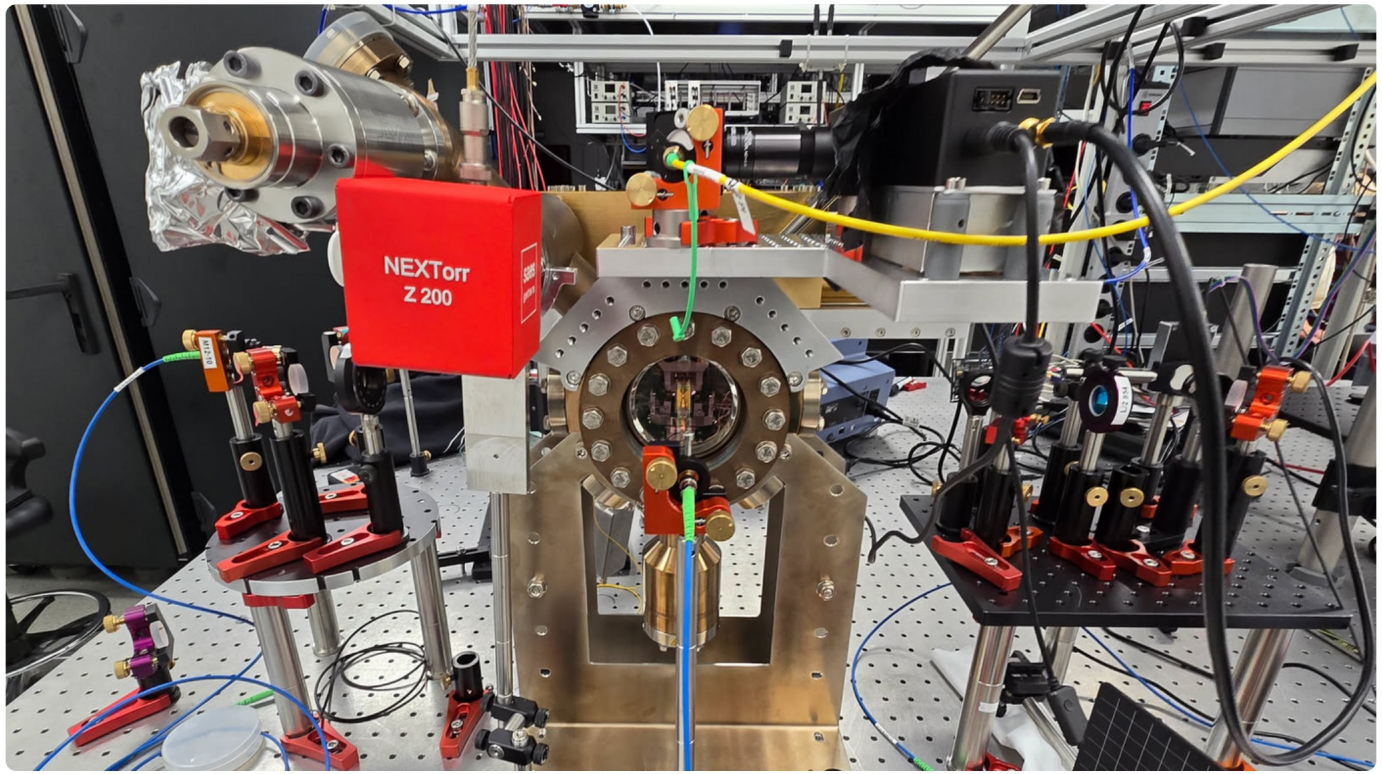
1. Deterministic **ion-photon entanglement**
2. **Quantum memory** for remote entanglement
3. **Entangled quantum nodes**
4. **Remote preparation** of stored multipartite entanglement

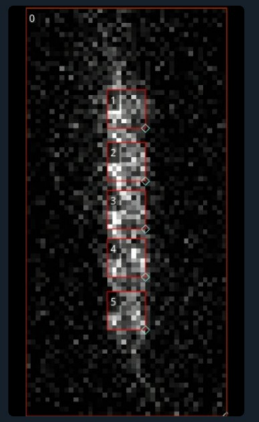
```
10 ions, 2 nodes
```

Distributed Quantum Systems

universität innsbruck

Current status





OCTOBER 2026

First ions trapped!

Cavity finesse ≈ 57,000

NEXT

Ion-photon entanglement

Real-time experiments

Ion-ion entanglement with **another network node!**

Take-home message

1 **Quantum networks for DQC can be catered to applications**

Scaling of connectivity will kill you!
We have shown that collective operations in one round lead to a significantly lower communication burden (e.g., QAOA, Grover)

2 **DQC emulators and machines need standard benchmarks**

We identified a test case for four emulators -> QFT is communication-dense

3 **Building the tools for DQC-HPC integration**

CUNQA raises vQPUs -> execution on QPU or emulator part of the HPC queue system
NetQIR: user defines DQC algorithm -> compiler and runtime pick protocol and route

4 **Datacentre design constraints**

Latency set by integration layer, not the QPU

NOW	NISQ algorithms on quantum hardware. Protocols emulated at scale on FinisTerrae III with CUNQA and NetQMPI
NEXT	Standardised benchmarking → informs hardware acquisition. Simulate noise on the communication directives
FUTURE	Small **DQC network deployed on-premises** → pooled QPUs as ordinary datacentre resources

THANK YOU! Questions?

ifllovo@cesga.es

PAPERS

Network-assisted collective operations
I. F. Llovo, et al., *IEEE Trans. Quantum Eng.* **6** 4101314 (2025)

Review of distributed quantum computing
D. Barral, et al., *Comput. Sci. Rev.* **57** 100747 (2025)

Benchmarking DQC emulators
G. Díaz-Camacho, et al., arXiv:2512.01807 (2025)

CUNQA
J. Vázquez-Pérez, et al., arXiv:2511.05209 (2025)

NetQIR and **NetQMPI**
F. J. Cardama, et al., *Future Gener. Comput. Syst.* **174** 107989 (2026)
F. J. Cardama, et al., *IEEE Access* **14** 125459 (2026)

QMIO and **interregional QKD**
J. Cacheiro, et al., arXiv:2505.19267 (2025)
D. Barral, et al., arXiv:2604.20376 (2026)

CODE

CUNQA
github.com/CESGA-Quantum-Spain/cunqa

NetQMPI
github.com/netqir/netqmpi

NetQIR spec, SDK and grammar
github.com/netqir

Emulator benchmarking
github.com/gdiazcamacho/DQC-Emulator-Benchmarking

Router protocols, QAOA and Grover
zenodo.org/records/17376198

KMSTN, key management across QKD domains
doi.org/10.5281/zenodo.20713366

XUNTA DE GALICIA
AXENCIA GALEGA DE INNOVACIÓN
GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
Plan de Recuperación, Transformación y Resiliencia
Funded by the European Union NextGenerationEU
Fondos Europeos
FWF

BACKUP SLIDES

A distributed Grover's search algorithm

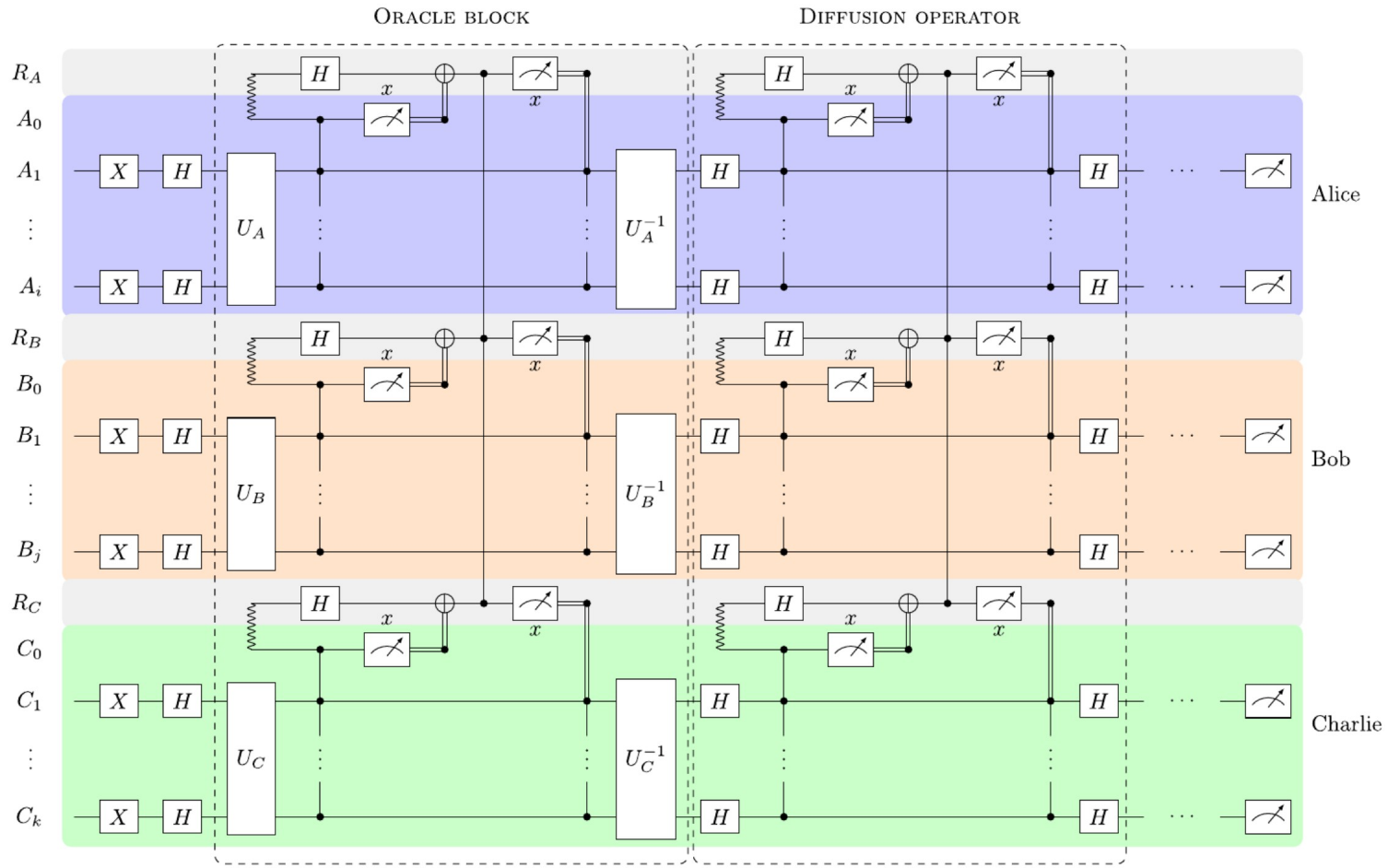
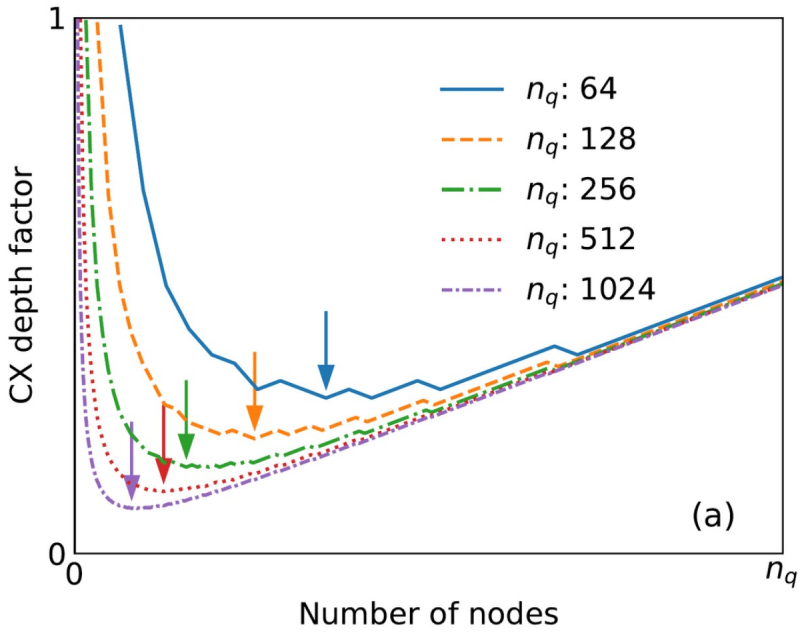


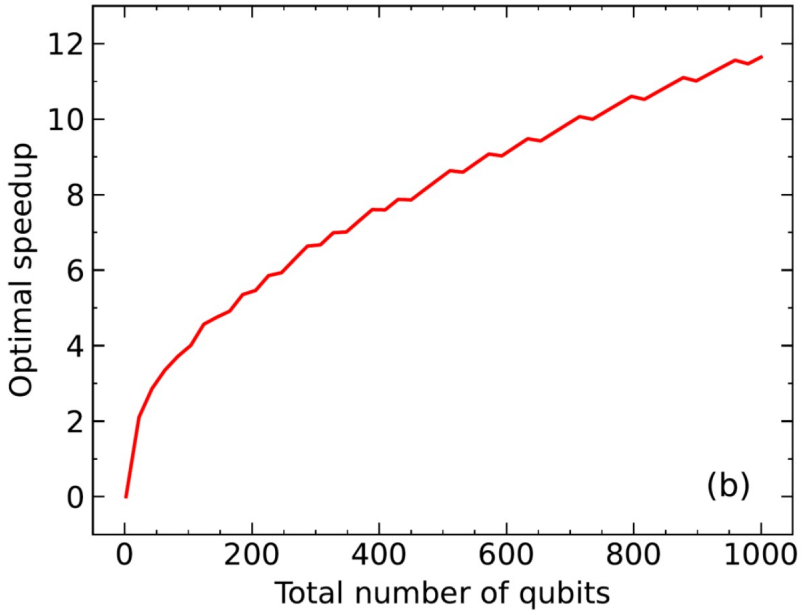
Figure 8 of I. F. Llovo, et al., *IEEE Trans. Quantum Eng.* **6** 4101314 (2025)

Optimising circuit distribution (Grover's algorithm)

Grover layer = 2 MCZ gates (for many oracles). Split an N-qubit MCZ over k nodes → CX depth **O(12N/k + 6k)**. Chunks run in parallel; the k-qubit gate in the router pushes back. **How many nodes should we use for maximum throughput?**



Depth against monolithic. Arrows: optimum



Speedup at optimal number of nodes for each system size

OPTIMAL SPLIT

$k = \sqrt{(2N)}$

nodes, N/k data qubits each

SPEEDUP THERE

$O(\sqrt{N})$

12× at 1024 qubits

COST

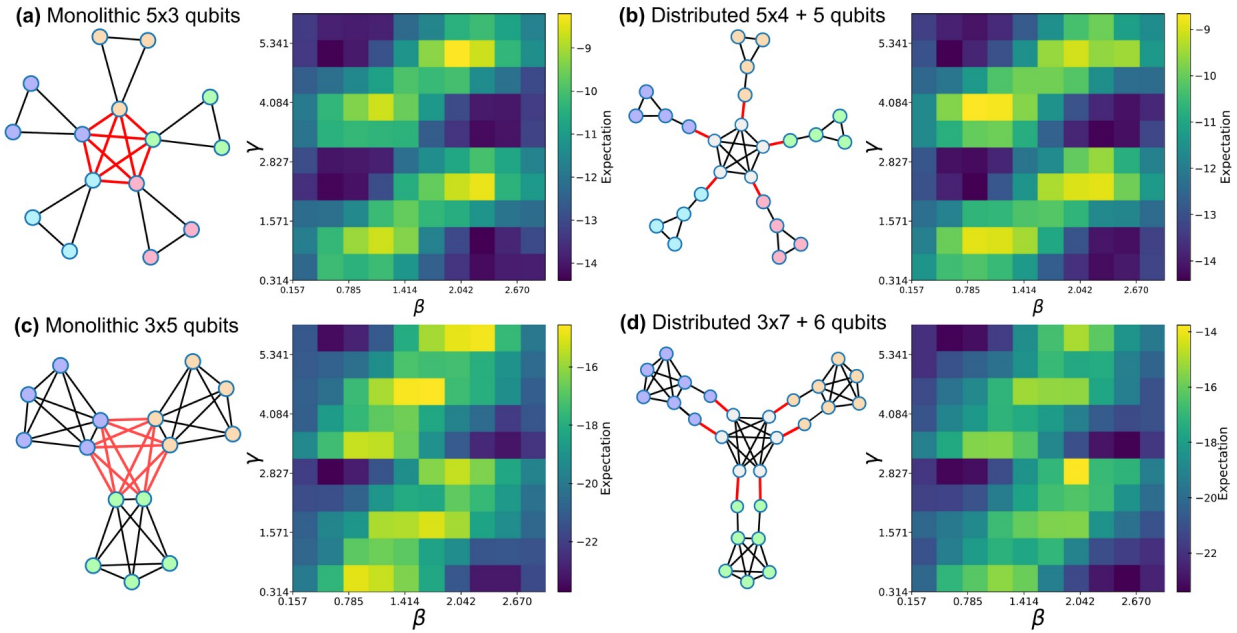
2k ebits

per Grover layer

I. F. Llovo, et al., *IEEE Trans. Quantum Eng.* **6** 4101314 (2025)

Collective operations instead of point-to-point

Commuting diagonal gates executed in a single fan-out → whole QAOA cost layer done in a single round



15-qubit MaxCut, one QAOA layer, 100 shots per point. (a, c) monolithic, (b, d) distributed through the router

Partitioning is star-like here → an unstructured graph may gain nothing

CROSS-QPU OPERATIONS

$O(k^2M^2) \rightarrow kM$

for k QPUs, M communication qubits each

NONLOCAL ROUNDS

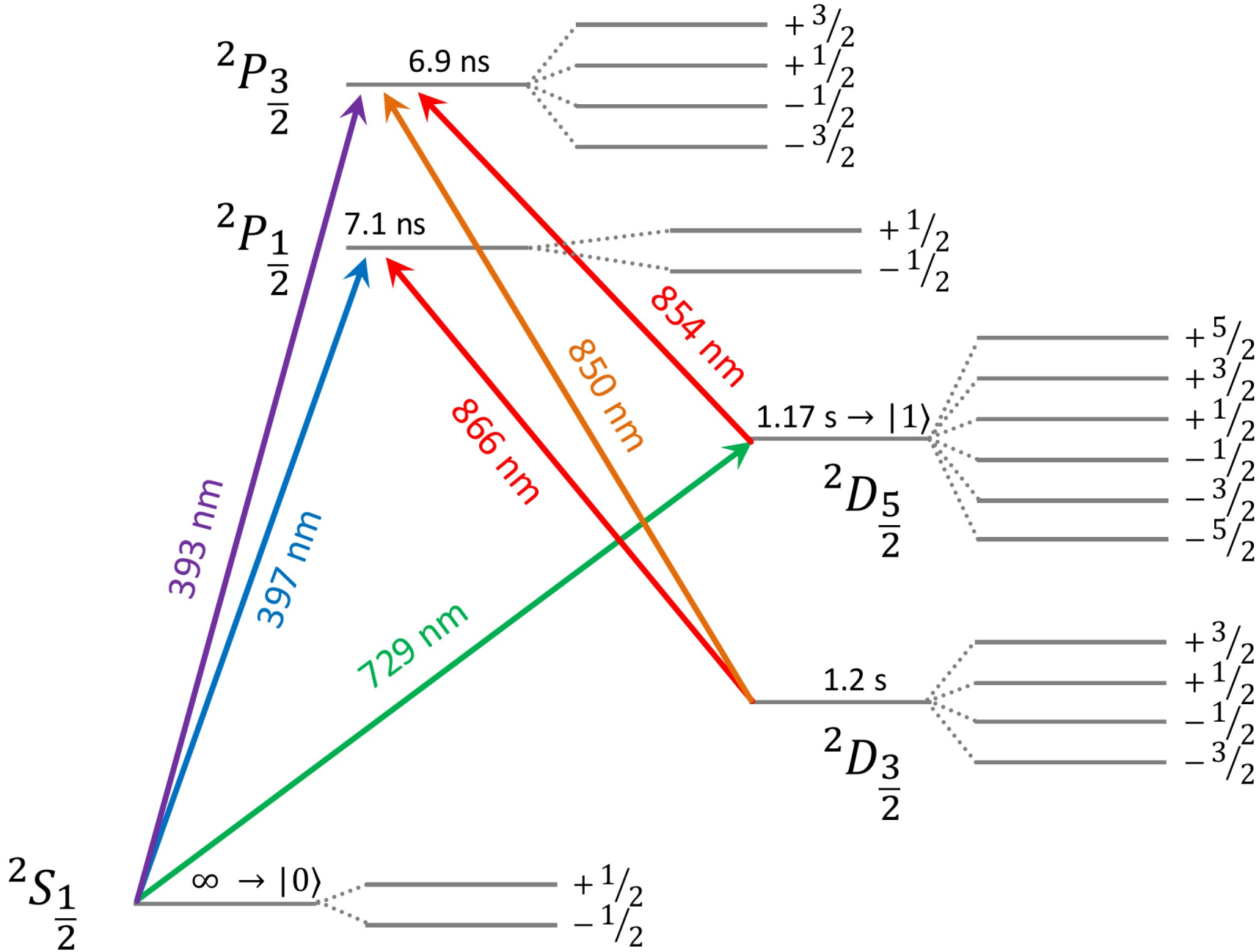
several → one

Swap-and-teleport needs a new round once cross-QPU gates outnumber communication qubits

CONNECTING SMALLER NODES

Smaller devices are easier to build and operate
Nodes join or leave by connecting to the router

I. F. Llovo, et al., *IEEE Trans. Quantum Eng.* **6** 4101314 (2025)


$^2P_{\frac{3}{2}}$
6.9 ns
$+3/2$
$+1/2$
$-1/2$
$-3/2$
$^2P_{\frac{1}{2}}$
7.1 ns
$+1/2$
$-1/2$
854 nm
850 nm
866 nm
393 nm
397 nm
729 nm
1.17 s → $|1\rangle$
$^2D_{\frac{5}{2}}$
$+5/2$
$+3/2$
$+1/2$
$-1/2$
$-3/2$
$-5/2$
1.2 s
$^2D_{\frac{3}{2}}$
$+3/2$
$+1/2$
$-1/2$
$-3/2$
$^2S_{\frac{1}{2}}$
∞ → $|0\rangle$
$+1/2$
$-1/2$

Getting DQC into the datacentre

NOW	NISQ algorithms on quantum hardware Integration tools for HPC workflows Protocols emulated at scale on FinisTerrae III with CUNQA and NetQMPI
NEXT	Standardised benchmarking → informs hardware acquisition DQC-HPC integration tools Simulate noise on the communication directives
FUTURE	Small **DQC network deployed on-premises** → pooled QPUs as ordinary datacentre resources