Arqon suite of quantum network control applications

Gauthier, Scarlett; Beauchamp, Thomas R.; Wehner, Stephanie

DOI
10.1145/3749096.3750035

Publication date
2025

Document Version
Final published version

Published in
QuNet 2025 - Proceedings of the 2nd Workshop on Quantum Networks and Distributed Quantum Computing, Part of SIGCOMM 2025

Citation (APA)
Gauthier, S., Beauchamp, T. R., & Wehner, S. (2025). Arqon suite of quantum network control applications. In V. Martin, N. Yu, & C. Qian (Eds.), *QuNet 2025 - Proceedings of the 2nd Workshop on Quantum Networks and Distributed Quantum Computing, Part of SIGCOMM 2025* (pp. 49-52). (QuNet 2025 - Proceedings of the 2nd Workshop on Quantum Networks and Distributed Quantum Computing, Part of SIGCOMM 2025). Association for Computing Machinery (ACM). https://doi.org/10.1145/3749096.3750035

Arqon suite of quantum network control applications

Extended Abstract

Scarlett Gauthier[*]
QuTech
Delft University of Technology
Delft, The Netherlands
s.s.gauthier@tudelft.nl

Thomas R. Beauchamp[†]
QuTech
Delft University of Technology
Delft, The Netherlands
t.r.beauchamp@tudelft.nl

Stephanie Wehner
QuTech
Delft University of Technology
Delft, The Netherlands
s.d.c.wehner@tudelft.nl

ABSTRACT

The aim of a quantum network is to enable users to successfully execute applications on their quantum end nodes. Users of mature networks, such as the internet, the postal network, or the telephone network expect their demands for service to be satisfied reliably. Here, we present an extended abstract introducing *Arqon*, a suite of control applications capable of delivering reliable service to end nodes. We define a full set of reliability requirements and demonstrate through a numeric evaluation that Arqon is capable of simultaneously satisfying all requirements.

CCS CONCEPTS

• **Networks** → **Network architectures**; *Network simulations*; *Network design principles*; • **Hardware** → **Quantum technologies**;

1 INTRODUCTION

To enable the execution of arbitrary quantum network applications [2, 5, 8, 9, 13, 14, 16, 18, 21], a network must provide the service of supplying or mediating the generation of packets of entanglement between end nodes [4, 10]. Quantum processing end nodes controlled by the state of the art operating system [11, 22] require a network schedule dictating when they can access internal resources of the network, to attempt end-to-end entangled link generation. To coordinate construction and distribution of such network schedules, a modular, centrally controlled quantum network architecture comprising five distinct control applications was proposed in [4]. In a proof of principle implementation and evaluation, the proportion of satisfied demands decreased substantially as the number submitted increased [4], exposing a need for improved control applications. More broadly, an outstanding challenge in the design of quantum network architectures is to develop control applications that can deliver reliable service to end nodes.

In computer networking, a system or protocol is reliable if it is able to consistently deliver services to their intended recipients without error or significant delay [19]. Adapting this principle to the service of a quantum network, we define the essential features of reliable service to be:

1. *that demands receive accept/reject responses, making it clear to end nodes whether or not they can expect service;*
2. *that accepted demands are satisfied with a tolerably high probability;*
3. *that demands are satisfied before their deadlines;*
4. *and that the amount or frequency of demands from other users should not disrupt service to accepted demands.*

Arqon is a suite of control applications for the network architecture in [4] which is designed to overcome the challenge of delivering reliable service to end nodes. The control applications *Demand Registration* and *Admission Control* filter demands, determining which to accept for service. The *Compute Schedule* and *Distribute Schedule* applications periodically create and distribute network schedules covering a duration of time T^{SI}, called the scheduling interval. Finally, the *Network Capabilities Manager* maintains a database of the achievable rates and fidelities for all possible end-to-end links. As in [4], end nodes submit demands for packets of entanglement generation to the network in the format

$$d = (\mathfrak{p}; t^{\text{minsep}}, t^{\text{expiry}}, N^{\text{inst}}), \tag{1}$$

where $\mathfrak{p} = (w, s, F)$ defines a packet of s pairs to be generated within time window w, with minimum fidelity F; t^{minsep} is a minimum time separation between packets, t^{expiry} is the expiry time, and N^{inst} is the number of successes required to satisfy the demand. Network schedules allocate *packet generation attempts* (PGAs) to accepted demands. A PGA for

---

[*] These authors contributed equally

[†] These authors contributed equally

QuNet '25, September 8–11, 2025, Coimbra, Portugal

ACM ISBN 979-8-4007-2097-0/2025/09.
https://doi.org/10.1145/3749096.3750035

demand d has duration E_d, success probability p_d^{packet}, and reserves internal resources on a route through the network.

2 CONTRIBUTION HIGHLIGHTS

The first reliability requirement is implemented by Arqon's *Demand Registration* and *Admission Control* applications. To implement the second requirement, each accepted demand d is associated with a maximum probability of not being satisfied, $\epsilon_d^{\text{service}}$. If a demand receives *minimal allocation* (Definition 2.1) in every distributed schedule, it will be satisfied with probability at least $(1-\epsilon_d^{\text{service}})$. Requirement three implies fast schedule computation must be a design criterion for Arqon's *Compute Schedule* application. A scheduling algorithm may be considered fast if it's operational complexity (the complexity of the algorithm implementation), is polynomial in N, the number of accepted demands to be scheduled.

Definition 2.1 (Minimal allocation). *Let t^{start} be the start time of the first scheduling interval in which demand d may be scheduled, and let $n^{SI} = \lceil (t^{expiry} - t^{start})/T^{SI} \rceil$. Let $\mathcal{N}_d^{SI}$ be such that if $X \sim \text{Binomial}(\mathcal{N}_d^{SI} n^{SI}, p_d^{packet})$ then $\mathbb{P}[X < N_d^{inst}] < \epsilon_d^{service}$. Let $\#_d(S)$ be the number of PGAs to serve demand d scheduled in schedule S. Then we say a demand receives **minimal allocation** in schedule S if $\#_d(S) >= \mathcal{N}_d^{SI}$.*

A challenge addressed by Arqon's *Admission Control* application is to accurately account for the time required by accepted demands. Accurate accounting is a method that can enable meeting the fourth reliability requirement. This is primarily challenging due to the t^{minsep} parameter, which is not typical in resource allocation problems. Let ϕ be the set of M indices of previously accepted, but not yet expired, demands. In this extended abstract the simplifying assumption that $\mathcal{N}_d^{SI} = \mathcal{N}^{SI}\ \forall\ d \in \phi$ is included. In the full version of this work, calculation of the time required by accepted demands is generalized. The amount of *time required* in each scheduling interval to satisfy demand set ϕ is given by $R(\phi)$,

$$R(\phi) = \mathcal{N}^{SI} \cdot \max\left(\max_{d\in\phi}(E_d + t_d^{\text{minsep}}), \sum_{d=0}^{M} E_d\right). \tag{2}$$

The core functionality of *Admission Control* can be captured by a *simplified admission control test* which rules that a demand with index $M+1$ can be accepted if

$$R\big(\phi \cup \{M+1\}\big) \leq T^{SI}. \tag{3}$$

The *Compute Schedule* application is a custom two-phase scheduling algorithm based on modifications of Round-Robin [20]. In the *Direct Allocation* phase, the minimal allocation of each demand is added to the network schedule. Only the *Direct Allocation* phase of the algorithm is time critical, as it must be completed on time to ensure the second and third reliability requirements are met. The *Bonus Allocation* phase sequentially attempts to fill any gaps in the network schedule with PGAs of duration less than or equal the gap duration.

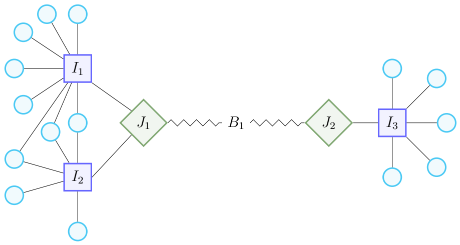

Figure 1: Quantum network topology used in evaluations. I_1, I_2, I_3 are entanglement generation interfaces such as Bell-state analyzers or entangled photon sources [7]. J_1, J_2 are junction nodes equipped with one or more quantum memories, which act as border nodes for a repeater chain backbone, $B1$.

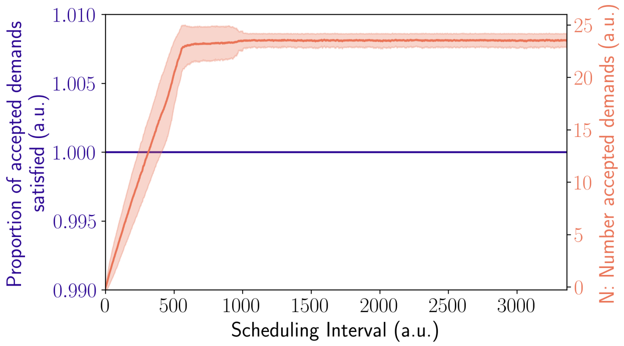

Figure 2: Proportion of accepted demands which are satisfied (left, blue) and the total number of accepted demands (right, orange), compared to the scheduling interval. Scheduling interval duration was 30 minutes. Shaded regions show 1 standard deviation of the mean.

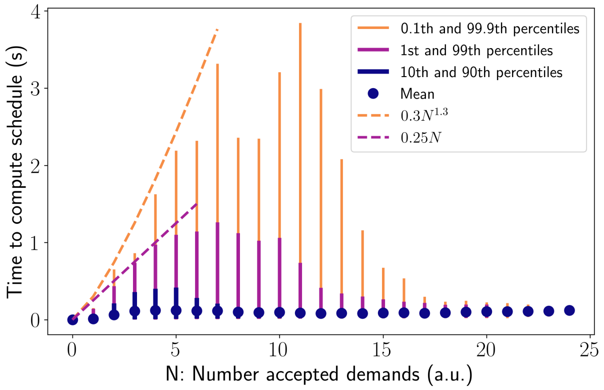

Figure 3: Time required to compute the network schedule as compared to N, the number of accepted demands. Dashed lines for $0.25N$ and $0.3N^{1.3}$ bound the rising edge of the 99^{th} and 99.9^{th} percentiles, respectively.

3 EVALUATION

We evaluate the performance of python implementation of Arqon using a simulated long-distance quantum network with the topology indicated in Figure 1. In total, 53 of the $\binom{15}{2} = 105$ unique pairs of end nodes repeatedly submitted demands for Quantum Key Distribution (QKD) [1, 5, 14], Blind Quantum Computing (BQC) [2, 8, 9], or either type of application with 1-2 rounds of entanglement purification [6, 12]. Realistic minimum fidelity requirements were used in the demands, calculated according to [1, 3, 10, 12, 17]. A simulated *Network Capabilities Manager* associated a maximum rate of successful end-to-end link generation and corresponding link fidelity with each route through the network. A database specifying the network capabilities and topology used in these evaluations is included as a supplement to this work [15]. Demands were rejected by the *Demand Registration* application if the required fidelity was higher than the maximum possible for any route connecting the end nodes of the demand. The proportion of accepted demands satisfied is the relevant performance metric for evaluating Arqon against our reliability requirements. An accepted demand is satisfied if both a) a number of PGAs $\geq N^{SI}$ of the demand are scheduled in every scheduling interval before the demand expires or is terminated by the end nodes and b) a number of packets $\geq N^{\text{inst}}$ are successfully generated by the end nodes before t^{expiry}. Confirming reliability requirements (2-4), Figure 2 demonstrates that over the course of the 10 weeks of continuous network operation simulated, the proportion of accepted demands which were satisfied was always 1.0, even as the average number of accepted demands (N) increased over the scheduling intervals. Figure 3 demonstrates that the *Compute Schedule* application achieves fast schedule computation, where the time required to compute the schedule grows polynomially in N, with $0.3N^{1.3}$ bounding 99.9% of observed schedule computation times.

ACKNOWLEDGMENTS

This work was supported by the Quantum Internet Alliance (QIA). QIA has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101102140. SW also acknowledges funding from NWO VICI.

REFERENCES

[1] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß. 2013. Quantum repeaters and quantum key distribution: Analysis of secret-key rates. *Physical Review A* 87, 5 (May 2013), 052315. https://doi.org/10.1103/physreva.87.052315

[2] P. Arright and L. Salvail. 2006. Blind Quantum Computation. *International Journal of Quantum Information* 04, 05 (2006), 883–898.

[3] G. Avis, F. Ferreira da Silva, T. Coopmans, A. Dahlberg, H. Jirovská, D. Maier, J. Rabbie, A. Torres-Knoop, and S. Wehner. 2023. Requirements for a processing-node quantum repeater on a real-world fiber grid. *npj Quantum Information* 9, 1 (Oct. 2023). https://doi.org/10.1038/s41534-023-00765-x

[4] T. R. Beauchamp, H. Jirovská, S. Gauthier, and S. Wehner. 2025. A Modular Quantum Network Architecture for Integrating Network Scheduling with Local Program Execution. (2025). arXiv:quant-ph/2503.12582 https://arxiv.org/abs/2503.12582

[5] C. H. Bennett and G. Brassard. 2014. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science* 560 (Dec. 2014), 7–11. https://doi.org/10.1016/j.tcs.2014.05.025

[6] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters. 1996. Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels. *Physical Review Letters* 76, 5 (Jan. 1996), 722–725. https://doi.org/10.1103/physrevlett.76.722

[7] H.K.C. Beukers, M. Pasini, H. Choi, D. Englund, R. Hanson, and J. Borregaard. 2024. Remote-Entanglement Protocols for Stationary Qubits with Photonic Interfaces. *PRX Quantum* 5, 1 (March 2024), 010202. https://doi.org/10.1103/prxquantum.5.010202

[8] A. Broadbent, J. Fitzsimons, and E. Kashefi. 2009. Universal Blind Quantum Computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, Atlanta, Georgia, USA, 517–526. https://doi.org/10.1109/focs.2009.36

[9] A. M. Childs. 2005. Secure assisted quantum computation. *Quantum Information and Computation* 5, 6 (2005), 456–466. https://doi.org/10.26421/QIC5.6 arXiv:quant-ph/0111046.

[10] B. Davies, T. Beauchamp, G. Vardoyan, and S. Wehner. 2024. Tools for the Analysis of Quantum Protocols Requiring State Generation Within a Time Window. *IEEE Transactions on Quantum Engineering* 5 (2024), 1–20. https://doi.org/10.1109/TQE.2024.3358674

[11] C. Delle Donne, M. Iuliano, B. van der Vecht, G. M. Ferreira, H. Jirovská, T. J. W. van der Steenhoven, A. Dahlberg, M. Skrzypczyk, D. Fioretto, M. Teller, P. Filippov, A. R.-P. Montblanch, J. Fischer, H. B. van Ommen, N. Demetriou, D. Leichtle, L. Music, H. Ollivier, I. te Raa, W. Kozlowski, T. H. Taminiau, P. Pawełczak, T. E. Northup, R. Hanson, and S. Wehner. 2025. An operating system for executing applications on quantum network nodes. *Nature* 639, 8054 (March 2025), 321–328. https://doi.org/10.1038/s41586-025-08704-w

[12] D. Deutsch, A. Ekert, C. Jozsa, R.and Macchiavello, S. Popescu, and A. Sanpera. 1996. Quantum Privacy Amplification and the Security of Quantum Cryptography over Noisy Channels. *Physical Review Letters* 77, 13 (Sept. 1996), 2818–2821. https://doi.org/10.1103/physrevlett.77.2818

[13] D. Ding and L. Jiang. 2024. Coordinating Decisions via Quantum Telepathy. (Sept. 2024). https://doi.org/10.48550/arXiv.2407.21723 arXiv:2407.21723 [quant-ph].

[14] A. K. Ekert. 1991. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* 67 (Aug. 1991), 661–663. Issue 6. https://doi.org/10.1103/PhysRevLett.67.661

[15] S. Gauthier, T. R. Beauchamp, and S. Wehner. 2025. Database of network capabilities and network toplogy used in dumbbell topology evaluation of Arqon: dumbbellconfig.db. (2025). See supplement.

[16] M. Hasanpour, S. Shariat, P. Barnaghi, S. A. Hoseinitabatabaei, S. Vahid, and R. Tafazolli. 2017. Quantum load balancing in ad hoc networks. *Quantum Information Processing* 16, 6 (April 2017), 148. https://doi.org/10.1007/s11128-017-1578-y

[17] M. Horodecki, P. Horodecki, and R. Horodecki. 1999. General teleportation channel, singlet fraction, and quasidistillation. *Phys. Rev. A* 60 (Sep 1999), 1888–1898. Issue 3. https://doi.org/10.1103/PhysRevA.60.1888

[18] P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin. 2014. A quantum network of clocks. *Nature Physics* 10, 8 (June 2014), 582–587. https://doi.org/10.1038/nphys3000

[19] L. Peterson and B. Davie. 2012. *Computer Networks: A Systems Approach*. Elsevier, Amsterdam, The Netherlands. https://book.systemsapproach.org/ Open-source edition available at https://book.systemsapproach.org/, licensed under CC BY 4.0.

[20] A. S. Tanenbaum. 2001. *Modern Operating Systems* (second ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[21] S. Tani, H. Kobayashi, and K. Matsumoto. 2012. Exact Quantum Algorithms for the Leader Election Problem. *ACM Transactions on Computation Theory* 4, 1 (March 2012), 1–24. https://doi.org/10.1145/2141938.2141939

[22] B. van der Vecht, A. Talay Yücel, H. Jirovská, and S. Wehner. 2025. Qoala: an Application Execution Environment for Quantum Internet Nodes. (2025). https://doi.org/10.48550/arXiv.2502.17296 arXiv:2502.17296 [quant-ph] arXiv:2502.17296 [quant-ph].