Protocol Design for Quantum Repeater Networks

Luciano Aparicio
Graduate School of Information
Science and Technology
The University of Tokyo, Japan
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
+81-3-5841-7465
lucho@hongo.wide.ad.jp

Rodney Van Meter
Faculty of Environment
and Information Studies
Keio University, Japan
5322, Endo, Fujisawa,
Kanagawa, 252-0882 Japan
+81-4-6649-1100
rdv@sfc.wide.ad.jp

Hiroshi Esaki
Graduate School of Information
Science and Technology
The University of Tokyo, Japan
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
+81-3-5841-7465
hiroshi@wide.ad.jp

ABSTRACT

When built, quantum repeater networks will require classical network protocols to control the quantum operations. However, existing work on repeaters has focused on the quantum operations themselves, with less attention paid to the contents, semantics, ordering and reliability of the classical control messages. In this paper we define and describe our implementation of the classical control protocols. The state machines and packet sequences for the three protocol layers are presented, and operation confirmed by running the protocols over simulations of the physical network. We also show that proper management of the resources in a bottleneck link allows the aggregate throughput of two end-to-end flows to substantially exceed that of a single flow. Our layered architectural framework will support independent evolution of the separate protocol layers.

Categories and Subject Descriptors

C.2.1. [**Computer Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer Communication Networks**]: Protocol Architecture

General Terms

Design, Theory.

Keywords

Quantum communication, Quantum repeater, Quantum networks

1. INTRODUCTION

Applications that use distributed quantum properties, such as QKD (Quantum Key Distribution) [14, 12], quantum Byzantine agreement [2], possibly improved optical interferometers for telescopes [15], and other forms of distributed computation [8, 3], have the limitation that the *fidelity* of quantum states and the probability of success decrease with distance, making the use of these systems over long distances almost impossible. Therefore, researchers have proposed the design of quantum repeater networks [11] which would maintain *distributed quantum states* across greater distances.

Networks of quantum repeaters utilize three concepts (explained in more detail in Section 2) to execute a distributed algorithm that creates *entangled* quantum states between nodes that are far apart: a basic *entanglement mechanism* which depends on the physical implementation, error management (in this work, we study a method known as *purification*), and finally a quantum state propagation layer (here we implement *entanglement swapping*, which builds multi-hop connections from single-hop connections). Some researchers are investigating approaches that are substantially different from entanglement swapping [18, 16, 13]. Here we focus on swapping, but the layered architecture approach is broadly applicable, allowing other implementations to replace only a single layer in the protocol stack.

Previous work primarily focused on the physical and mathematical tools for building repeaters. Classical information is also needed to enable *teleportation* and swapping, as many quantum operations are not deterministic, and results of quantum measurements need to be reported to distant partners before further operations can proceed. Also, operations in the middle of the network must be coordinated to route and swap properly. This requires classical messages to make operations robust, but message propagation times penalize performance. Even though this delay is usually included in repeater simulations, prior work has not defined the protocols in detail, especially with respect to how all of the nodes make consistent decisions in a timely fashion.

In this work, we introduce a protocol stack for networks of quantum repeaters that considers all the necessary clas-

AINTEC'11, November 9–11, 2011, Bangkok, Thailand.

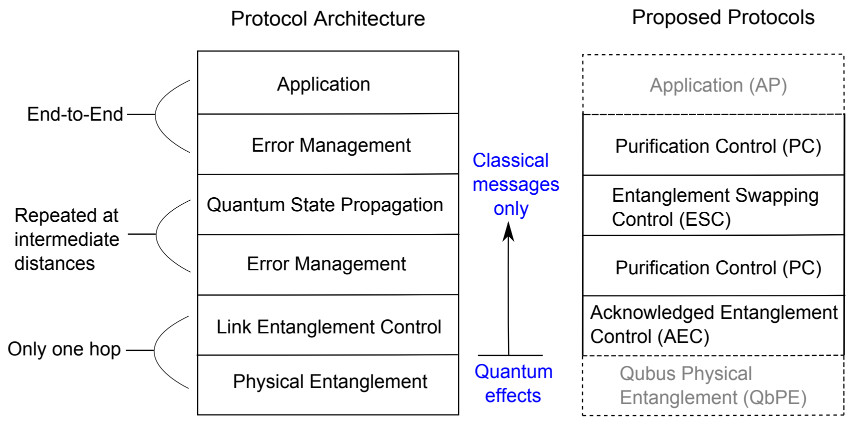
Protocol Architecture
Proposed Protocols
End-to-End
Repeated at intermediate distances
Only one hop
Application
Error Management
Quantum State Propagation
Error Management
Link Entanglement Control
Physical Entanglement
Classical messages only
Quantum effects
Application (AP)
Purification Control (PC)
Entanglement Swapping Control (ESC)
Purification Control (PC)
Acknowledged Entanglement Control (AEC)
Qubus Physical Entanglement (QbPE)

Figure 1: Protocol Stack Architecture and Proposed Protocols

sical messages and which can be easily adapted for different approaches at all three protocol layers. We simulated a qubus mechanism as the physical protocol, Deutsch purification as the protocol for error management, and entanglement swapping as the protocol responsible for making the entanglement span from end to end, in order to increase our confidence in the behavior of our network protocols. By adjusting the fidelity thresholds required for entanglement swapping, we show that some configurations boost the aggregate throughput for multiple flows significantly above the maximum for a single flow, taking advantage of resources that would otherwise sit idle. The operation of such complex networks and such delicate tuning of the system without formal protocol definitions would not be possible.

In Section 2, we introduce some concepts of quantum information science that are used in this work. In Section 3, we begin by describing the division of work, separating some of the functions of a quantum repeater into protocol layers that exchange messages with a partner. We also explain the finite state machines that control individual qubits or Bell pairs in a distributed fashion. Section 4 presents the results of our simulations of the protocol. We conclude with future work and conclusions in Section 5.

2. BACKGROUND

Communication of quantum states depends on several quantum operations and properties: key among these is *entanglement*, in which the states of two or more quantum bits (qubits) are not independent. This operation is done by interacting qubits, producing this high correlation among their quantum states. For communications, one useful, basic form of entanglement is a *Bell pair*. Bell pairs can be created over a distance using optical pulses that are coupled to a qubit (represented as e.g. the spin of a single electron held in a quantum dot) at each end of a waveguide. Due to losses in the waveguide, this operation is probabilistic. Bell pairs can be used for teleporting a qubit from one location to another. The Bell pair is consumed in the process, so we must continually refresh the supply of available pairs. To cover distances of more than one hop, a form of teleportation called *entanglement swapping* is used to splice two short Bell pairs into one long one.

The fidelity of a quantum state describes how accurately the state matches our desired one; $F = 1.0$ indicates that the state is perfect. After entanglement succeeds, usually the fidelity is not high enough for distributed quantum computation, and entanglement swapping and memory decoherence further degrade the fidelity. Purification is an algorithm which boosts the fidelity of a Bell pair by sacrificing a second pair. As purification is a non-deterministic operation, many resources are needed in the process to obtain a high fidelity Bell pair.

Previous work on quantum repeaters [7, 9, 4, 5, 10] has proposed different ways to produce entanglement via single photons or via very weak laser pulses. These produce high-fidelity Bell pairs, which makes purification almost unnecessary, but with a low probability of success. Other approaches improve the probability of success at the cost of reducing the initial fidelity [19]. The qubus mechanism used in this work is described in Section 3.1.

3. PROTOCOL DESIGN

The process of designing quantum networks is similar to designing classical networks, as they require detailed protocol designs, including finite states machines to control physical resources and track logical state. A layered protocol stack has previously been proposed [20]; here we provide detailed designs for the individual layers. We give a brief description of each and the functions which are simulated. Fig. 1 shows the layers of this protocol.

One of the key purposes of the classical protocols is to keep track of the fidelity of Bell pairs. The fidelity is the probability that we measure the right quantum state, which is due to non-deterministic quantum mechanics. Fidelity can only be estimated and not measured. Measurement of fidelity cannot be done directly, as it describes the probability of finding the system in the ”right” state. Moreover, measurement destroys the entanglement. Instead, fidelity is

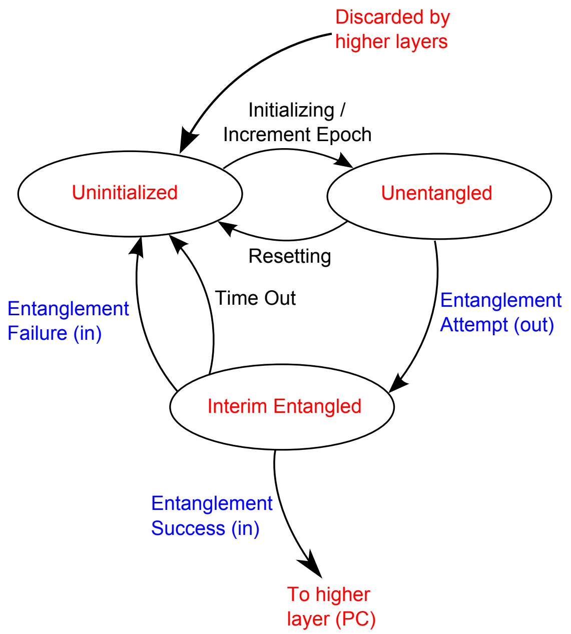
Discarded by higher layers
Initializing / Increment Epoch
Uninitialized
Unentangled
Resetting
Time Out
Entanglement Failure (in)
Entanglement Attempt (out)
Interim Entangled
Entanglement Success (in)
To higher layer (PC)

Figure 2: Finite state machine for Transmitter's ACKed Entanglement Control

tracked statistically for a given environment, by sacrificing some of the created Bell pairs, measuring them and confirming the state, as part of ongoing link monitoring. Decisions are made based on these estimated values in order to allow Bell pairs to be swapped or sent to the Application layer for use. The control of a qubit (a single-qubit buffer) is passed from layer to layer until consumed by the Application layer or reinitialized to start over from the lowest layer.

3.1 Physical Entanglement Layer: Qubus

The physical entanglement layer represents the physical interaction that creates Bell pairs between two different stations. There are many proposals for this layer and at the moment no clear winner. Our simulations model the qubus mechanism [19] in which laser pulses of many photons generate low-fidelity Bell pairs with high probability. For a distance of 20km, over an optical fiber with a 0.17 dB/km loss, the probability of success of the entanglement is around 36%, with an initial fidelity of 0.633. This probability is due to the attenuation in the optical fiber, which has an exponential increase with distance. Thus, photons may be lost when traveling, or may not be detected at the receiver. Once the Bell pairs are produced, decoherence also decreases their fidelity as a function of time, as information leaks into the surrounding environment.

The physical capabilities of different physical layers vary. Some support only a single physical transceiver qubit, and so can support only a single outstanding entanglement attempt. Others support independent multiplexing of incoming light pulses to local qubits, which is done by a classical herald pulse (trigger) followed by one or more quantum pulses. Our simulations assume such a capability.

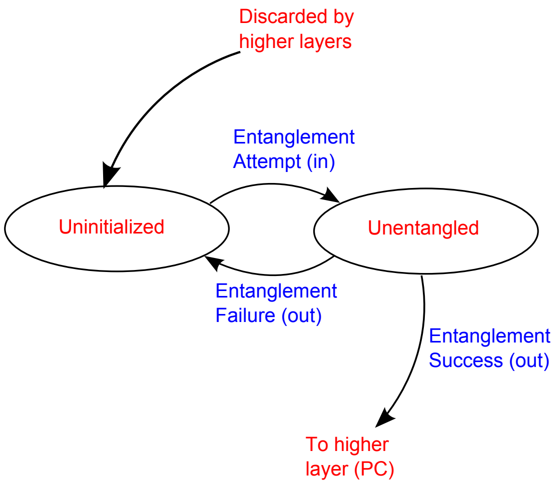
Discarded by higher layers
Entanglement Attempt (in)
Uninitialized
Unentangled
Entanglement Failure (out)
Entanglement Success (out)
To higher layer (PC)

Figure 3: Finite state machine for Receiver's ACKed Entanglement Control

3.2 Link Entanglement Control: ACKed Entanglement Control

The second layer, AEC (ACKed Entanglement Control), is responsible for managing the single-hop physical entanglement process, selecting qubits to attempt entanglement at each end of the link, and utilizing classical messages to report the results. When a laser pulse is detected by the receiver, measurements are done, and a message will be sent back to the transmitter, informing it which qubits on the receiver were entangled to which qubits on the transmitter. The stationary qubits in a repeater are not destroyed when the qubit is measured or reinitialized, though the quantum information held in the qubit is. Each qubit has an associated *epoch*, a counter of the number of times it has been initialized, to prevent old messages from being misinterpreted.

Once entanglement succeeds, this layer will transfer control of the Bell pair to a higher protocol layer.

Fig. 2 shows the finite state machine which describes the behavior of this layer for qubits in the transmitter, and Fig. 3 for qubits in the receiver.

Transitions represented in blue include an interaction with a remote station. OUT refers to messages sent, and IN to messages received. Black transitions represent local operations. The different states are:

- **Uninitialized**. This can be reached from a higher protocol layer or after starting up the repeater. Qubits are in an unknown state.

- **Unentangled**. During initialization of the qubits, they are prepared to have a known quantum state, ready to start entanglement, and the epoch is incremented. If the qubit's fidelity falls below a threshold, it becomes unusable and will be returned to the Uninitialized state.

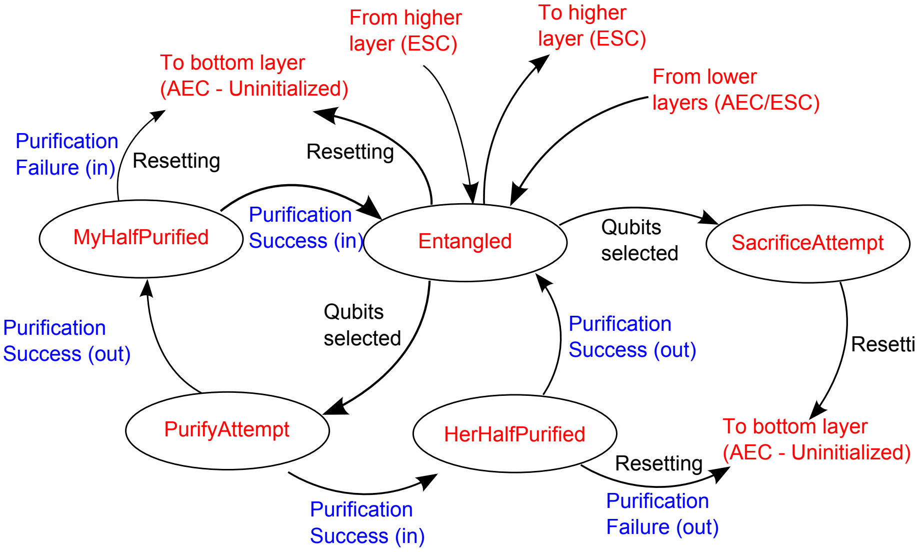

Figure 4: Finite state machine for Purification Control

- **Interim Entangled**. This state is reached after entanglement is attempted. This operation is done by sending a laser pulse to the remote station after interacting with the local qubits. The qubit will remain in this state until it receives an answer from the remote station or a local timer times out. This timer is based on the lifetime of the quantum memory used and in practice will be set to limit the impact on the fidelity to around 1%. If an Entanglement Failure message is received or no answer is received before the timer expires, the qubit will be moved to Uninitialized to start over again. If an Entanglement Success message is received, the qubit will be moved to the next higher protocol layer, in this case, Purification Control.

3.3 Error Management Layer: Purification

In this work we use an error management method called purification. In order to purify a Bell pair (boost its fidelity), an additional Bell pair is sacrificed in the process. The third layer of the protocol, PC (Purification Control), is responsible for choosing two Bell pairs, and electing one pair to have its fidelity boosted and the other to be sacrificed, assuring that both stations make the same decisions. Purification is done between two arbitrary stations, and no other stations need to be considered. Thus, PC does not need to make any complex routing decisions, but it does need to be able to address any station in the network. After the PC layer confirms a sufficient fidelity for the Bell pairs, control is given to the next higher protocol layer, which could be the Application layer, or the Entanglement Swapping Control (ESC), depending on whether or not this round of purification was done between end-to-end stations. If the physical layer produces high-fidelity Bell pairs, there is no need to execute any purification, so this layer could be configured as a null layer.

In order to select qubits to purify, the possible values of fidelity are grouped into fidelity bands. Two qubits in the same band are chosen for purification. The node which starts the purification process will choose the qubit with the lowest memory address as the one to be purified, and the second one to sacrifice. The node which receives the purification attempt will make the same decisions based on the address of the qubits of the remote station, in this case the node which started the purification. This way assures that both nodes choose the same qubits for purification and for sacrifice.

Fig. 4 shows the finite state machine which describes the behavior of this layer. The states are:

- **Entangled**. This state indicates that the qubit is entangled to another qubit in a distant station, but with not enough fidelity to start teleportation. It can be entered from a lower layer like AEC (just after entanglement is produced) or ESC (as fidelity is always reduced by swapping), from a higher layer ESC (if decoherence affects the fidelity and the Bell pair needs to be purified again), or finally from the purification algorithm itself after successful purification. Once a high level of fidelity is reached, control of the qubit is transferred into the next higher layer (ESC).

  If the qubit remains in this state for a long time, the fidelity will drop due to decoherence. The qubit is moved to Uninitialized in order to attempt entanglement again.

- **PurifyAttempt**. Once we have two Bell pairs with similar fidelity, one Bell pair is assigned to this state. If purification succeeds, this pair will have its fidelity boosted. After attempting purification, the qubit is moved to the state MyHalfPurify or HerHalfPurify, depending on whether this station starts the purification before receiving any purification message from the remote station.

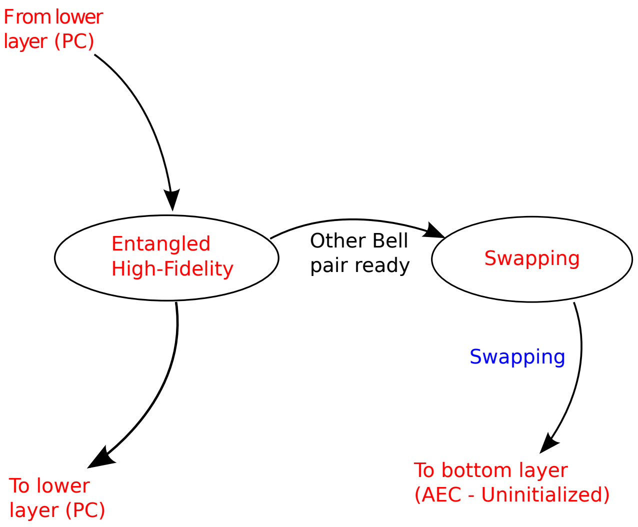

Figure 5: Finite State Machine for Entanglement Swapping Control for a middle node

- **SacrificeAttempt**. The second Bell pair chosen for purification is assigned to be sacrificed to improve the fidelity of the first Bell pair. Regardless of the result of the purification, this qubit will always be moved to Uninitialized state after the purification process finishes.

- **MyHalfPurify**. If the station starts the purification process, after sending a message to the remote station, the qubit is moved to this state, until it receives an answer. On success it will be moved to Entangled, or to Uninitialized on failure.

- **HerHalfPurify**. This state indicates that the station received a message notifying it that the remote station has started the purification process. If the operation on the local station succeeds, the qubit will be moved to Entangled, after sending a message to the remote station. If it fails, it will be moved to Uninitialized, after sending a Purify Failure message to the remote station.

3.4 Quantum State Propagation Layer: Entanglement Swapping Control

For networks which have more than two stations, further steps are required. As mentioned in the introduction, in this work we focus on *entanglement swapping*. A node in the middle of the network waits until it has two high-fidelity Bell pairs, one to each node it wants to couple. Then, this middle node performs the Bell state measurement that splices the two short Bell pairs into a longer one. As a consequence of this operation, the fidelity of the extended new Bell pair drops, and further purification may be necessary. ESC and PC are repeated until we have an end-to-end Bell pair of sufficient fidelity for our application.

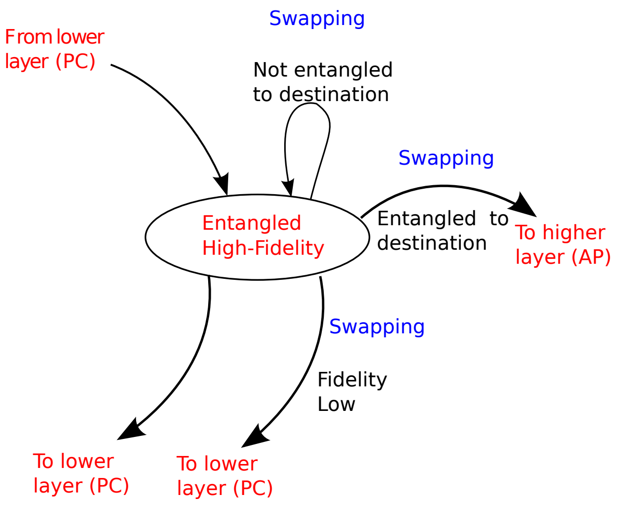

Figure 6: Finite state machine for Entanglement Swapping Control for an end node

This layer is responsible for administering the Bell pairs, especially for networks with shared resources. Important decisions, such as whether to purify or swap first, or when to swap, need to be carefully taken. The finite state machine for the stations that are in the middle and make the decision to swap is shown in Fig. 5. Finally, Fig. 6 shows the behavior of the end stations which received the swapping report from the middle station.

3.5 Application Layer

As of the time of writing, the most important, existing application is QKD (Quantum Key Distribution). A distributed quantum algorithm has also been proposed that will synchronize clocks to better-than-atomic-clock precision over a distance by using Bell pairs [17, 6]. In the future, if quantum computers are developed, distributed quantum computing will become important [14, 12, 2, 3, 8].

4. SIMULATIONS

Our simulator consists of 2000 lines of code written in C++ on top of Omnet++, a C++-based network simulator. The operation of a dumbbell quantum network was simulated for 10 seconds of simulated operation, taking around 150 seconds to complete on an Intel Core 2 T7200 CPU running at 2 GHz with 1 GB of RAM.

We simulated a qubus mechanism, with 20km hops, as described in Section 3.1. The number of qubits in each transmitter is 50, and 16 in the receivers. In all of our simulations, we use a target end-to-end fidelity of 0.98. We have run simulations for two cases: only one flow, and two flows competing for shared resources in the network shown in Fig. 7. Both flows are over three-hop paths (AEFB and CEFD), with the middle hop (EF) being a shared link and hence the throughput bottleneck. Bell pairs created on the EF link are assigned randomly to be used for the AB or CD flows. Used naively,

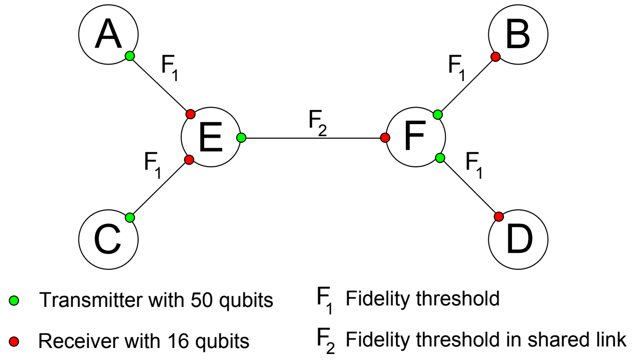
A
B
C
D
E
F
F_1
F_1
F_1
F_1
F_2
● Transmitter with 50 qubits
● Receiver with 16 qubits
F_1 Fidelity threshold
F_2 Fidelity threshold in shared link

Figure 7: Simulated dumbbell network

the first and third hops on each path will remain idle half of the time.

We hypothesized that careful tuning of the purification thresholds might better balance the system. If we raise the required fidelity on the under-utilized links, can we reduce the fidelity penalty incurred by entanglement swapping and improve aggregate performance? To test this hypothesis, we ran simulations with the purification threshold of each link set to several different values, while keeping the end-to-end target $F = 0.98$. We used two values for the first and third hops, $F_1 = 0.98$ and $F_1 = 0.99$, with the second case requiring an additional round of purification on those single hops. We then varied the fidelity threshold of the middle hop, $F_2 = 0.86, 0.94, 0.98$, altering the number of purification rounds required over that single hop. The aggregate throughput of the flows for each of the twelve simulations is shown in Fig. 8. With $F_1 = 0.99$, the performance of two flows doubled that of a single flow, as seen in the green and purple bars on the right of the figure. Fig. 9 plots the final, delivered fidelity of the same cases. In the $F_2 = 0.98$ cases, the tradeoff for higher throughput is that the final fidelity just barely clears our established end-to-end target of $F = 0.98$.

This gain in throughput occurs because making $F_1 = 0.99$ produces end-to-end Bell pairs with a fidelity above the fixed threshold just after the final swapping is done. Therefore, no additional end-to-end purification steps are required and these Bell pairs can be used immediately by the application layer, providing a throughput improvement. As no further purification is done, the end-to-end fidelity is lower than for the other cases, but it is high enough to be used.

In the lower performance cases, end-to-end Bell pairs do not have enough fidelity to be used by the application layer, so the network waits until another Bell pair is available in order to attempt purification with them. This clearly increases the final end-to-end fidelity of the Bell pair at a cost of throughput reduction.

5. FUTURE WORK AND CONCLUSIONS

We provide a layered protocol architecture for quantum repeater networks and the design of one protocol stack for purify-and-swap repeaters. Our stack consists of ACKed Entanglement Control, Purification Control, and Entanglement Swapping Control. The latter two can be combined recursively as necessary to span arbitrary numbers of hops. We intend to make the simulator open source, and we hope for our protocol implementations to be used in actual experiments with quantum repeaters.

We studied the behavior of a dumbbell network, applying statistical multiplexing to manage the shared resources. Careful tuning of the fidelity thresholds in this topology allowed us to utilize idle resources and effectively work around a bottleneck link. We obtained a higher total throughput of high-fidelity Bell pairs ($F \geq 0.98$) when two flows shared the network, compared to only one flow. In related work [1], we have tested these protocols in more complex topologies with more flows and found that the best multiplexing scheme was statistical multiplexing, which we use in this work.

For more complex topologies, routing in quantum networks should also be addressed as it will directly affect the quantum state propagation layer. Other remaining work includes optimization of the total throughput in these topologies with more flows competing for resources, and trying to find a relationship between the fidelity threshold, number of hops and the number of flows.

This proposed architecture can be used as a reference model much as the OSI model is for classical networks. Researchers are continuing to work on many different technologies for the physical layer and other approaches to error management and quantum state propagation, and we expect these advances to integrate smoothly with this layered architecture.

Acknowledgment

LA acknowledges funding from MEXT. This work was supported by JSPS KAKENHI 21500020.

6. REFERENCES

[1] L. Aparicio and R. Van Meter. Multiplexing schemes for quantum repeater networks. In *Proceedings SPIE*, volume 8163, page 816308, August 2011.

[2] M. Ben-Or and A. Hassidim. Fast quantum Byzantine agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 481–485. ACM, 2005.

[3] H. Buhrman and H. Röhrig. *Mathematical Foundations of Computer Science 2003*, chapter Distributed Quantum Computing, pages 1–20. Springer-Verlag, 2003.

[4] L. Childress, J. M. Taylor, A. S. Sørensen, and M. D. Lukin. Fault-tolerant quantum repeaters with minimal physical resources and implementations based on single-photon emitters. *Phys. Rev. A*, 72(5):052330, Nov 2005.

[5] C. W. Chou, H. de Riedmatten, D. Felinto, S. V. Polyakov, S. J. van Enk, and H. J. Kimble.

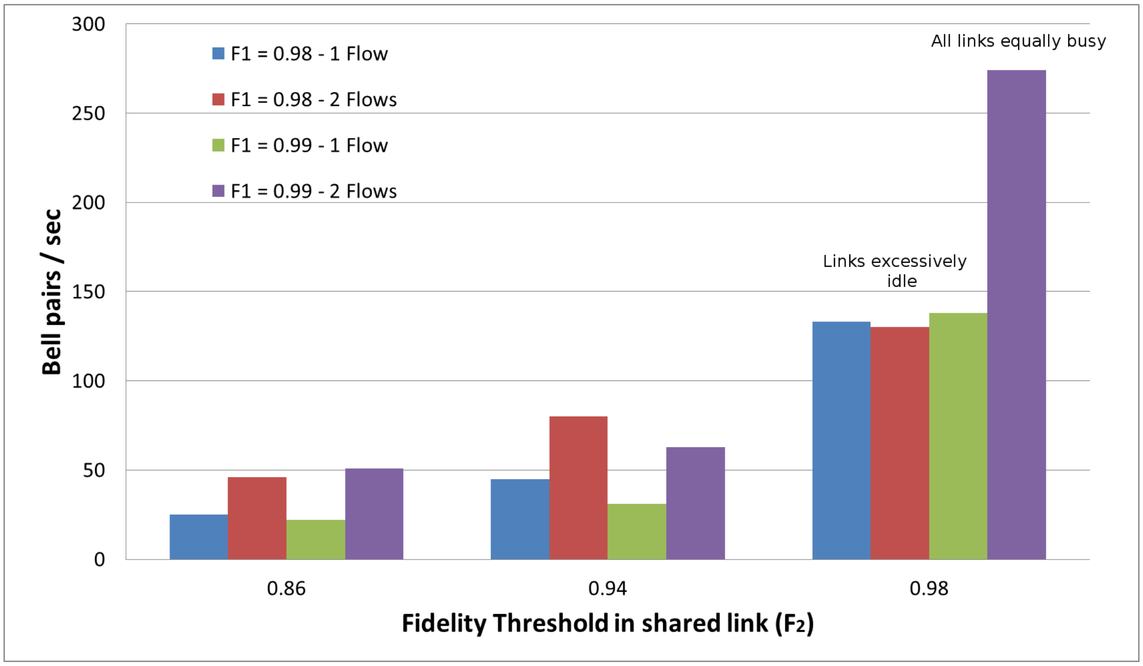

Figure 8: Throughput in Bell pairs/sec, one flow versus two flows

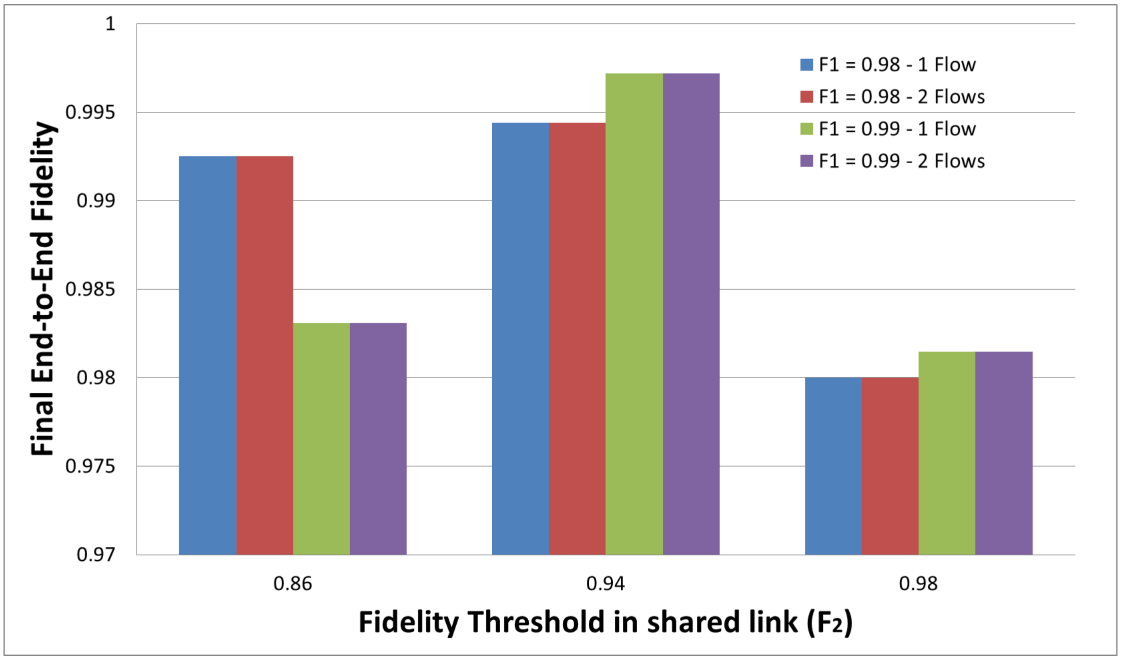

Figure 9: End-to-end fidelity of teleported qubits

Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature*, 438:828–832, Dec 2005.

[6] I. L. Chuang. Quantum algorithm for distributed clock synchronization. *Phys. Rev. Lett.*, 85(9):2006–2009, Aug 2000.

[7] J. I. Cirac, P. Zoller, H. J. Kimble, and H. Mabuchi. Quantum state transfer and entanglement distribution among distant nodes in a quantum network. *Phys. Rev. Lett.*, 78(16):3221–3224, Apr 1997.

[8] E. D'Hondt. *Distributed quantum computation: A measurement-based approach*. PhD thesis, Vrije Universiteit Brussel, July 2005.

[9] L. Duan, M. Lukin, J. Cirac, and P. Zoller. Long-distance quantum communication with atomic ensembles and linear optics. *Nature*, 414:413–418, 2001.

[10] L.-M. Duan and H. J. Kimble. Scalable photonic quantum computation through cavity-assisted interactions. *Phys. Rev. Lett.*, 92(12):127902, Mar 2004.

[11] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Phys. Rev. A*, 59(1):169–181, Jan 1999.

[12] C. Elliott, D. Pearson, and G. Troxel. Quantum cryptography in practice. In *Proc. SIGCOMM 2003*. ACM, ACM, Aug. 2003.

[13] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. L. Hollenberg. Surface code quantum communication. *Phys. Rev. Lett.*, 104(18):180503, May 2010.

[14] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden. Quantum cryptography. *Rev. Mod. Phys.*, 74(1):145–195, Mar 2002.

[15] D. Gottesman, T. Jennewein, and S. Croke. Longer-baseline telescopes using quantum repeaters. Arxiv preprint arXiv:1107.2939, 2011.

[16] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin. Quantum repeater with encoding. *Phys. Rev. A*, 79(3):032325, Mar 2009.

[17] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams. Quantum clock synchronization based on shared prior entanglement. *Phys. Rev. Lett.*, 85(9):2010–2013, Aug 2000.

[18] W. J. Munro, K. A. Harrison, A. M. Stephens, S. J. Devitt, and K. Nemoto. From quantum multiplexing to high-performance quantum networking. *Nature Photonics*, 2010.

[19] P. van Loock, T. D. Ladd, K. Sanaka, F. Yamaguchi, K. Nemoto, W. J. Munro, and Y. Yamamoto. Hybrid quantum repeater using bright coherent light. *Phys. Rev. Lett.*, 96(24):240501, Jun 2006.

[20] R. Van Meter, T. D. Ladd, W. J. Munro, and K. Nemoto. System design for a long-line quantum repeater. *IEEE/ACM Trans. Netw.*, 17(3):1002–1013, 2009.