Software-defined Quantum Network Switching

Travis S. Humble[1,2], Ronald J. Sadlier[1,2], Brian P. Williams[1], and Ryan C. Prout[1]

[1]Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, Tennessee 37831
[2]Bredesen Center, University of Tennessee, Knoxville, Tennessee, 37996

ABSTRACT

We present the design and implementation of a software-defined quantum networking protocol and software switch integrated with a numerical quantum channel simulator. Our protocol design leverages recent advances in the OpenFlow protocol that enable software-defined control and management of optical network traffic using side-channel metadata. We implement this design using customization of the open source vSwitch for optical network routing, and we test the implementation using a numerical simulator of the quantum channel alongside actual network traffic. Our results support the integration of quantum communication with existing optical transport methods.

Keywords: Quantum Networking, Software-Defined Networking, Modeling and Simulation

1. INTRODUCTION

The design and development of robust and extensible quantum networks is an important goal for future quantum information technology. There has been considerable theoretical and experimental effort surrounding the engineering of quantum key distribution (QKD) networks.[1–9] These have been heroic efforts to demonstrate possible quantum communication capabilities even though results from those systems, however, are often specific to QKD applications. While such systems represent fixed-point solutions and, in the broad sense of quantum network science, have limited functionality, they nonetheless emphasize the integration techniques need in quantum networks. However, there is an outstanding need for the development of robust network protocols and abstractions. As a near-term example, the on-going development of quantum repeater technologies highlights the need to modify even the most basic protocols for resource efficient quantum communication. These changes are also driven by the discovery of new protocols or use cases that cannot be foreseen during network construction. A longer term goal of establishing ad hoc quantum networks, which enable devices built from different technology bases to interface, will also require control and management methods that are not application or technology specific. The current variability in network components, operation, and behavior motivates our interest in building quantum networks that are programmable and can be modified quickly to take on new tasking.

We address the design of versatile quantum networks through the development of a programmable quantum switch. Our approach is based on the principles of software-defined networking (SDN), a concept that has emerged for managing next-generation conventional networks. Software-defined network switching allows for the forwarding rules to be configured remotely. OpenFlow is a communication protocol for programming SDN switches and configuring the switch flow tables that define packet forwarding. These principles separate the data plan that defines network traffic from the control plane that dictates how that traffic is routed. This allows us to reconfigure individual switches within a quantum network for purposes of accommodating new communication protocols.

Contact information: Travis S. Humble (humblets@ornl.gov)

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

The paper is organized as follows: after the introduction of Sec. 1, we describe the purpose and challenges of switching traffic in a quantum network in Sec. 2. In Sec. 3, we describe a software-defined protocol that manages quantum traffic switching using concurrent conventional packet switching protocols. We offer final remarks in Sec. 4.

2. QUANTUM NETWORK SWITCHING

The deployment of conventional networks has thrived with the scalable architectures and protocols underlying their design, and similar designs must be addressed in the development of quantum networks. In particular, conventional networks have prospered by isolating communication concerns into a layered representation, like the OSI model shown in Fig. 1. Many modern networks use packet switching to route traffic, in which a switch forwards packets based on source and destination addresses. A packet switch maintains an address table and uses forwarding rules based on the MAC and/or IP address information contained in the packet to determine the next destination. However, this approach cannot be used directly due to the consequences of the no-broadcast theorem,[10] the transmission of arbitrary quantum states through a multi-node infrastructure is significantly more challenging than represented by modern network transport protocols. Additional challenges also arise from consideration of the sensitivity of transmission schemes to physical noise processes as well as the complexity of synchronizing multiparty entanglement across the distributed nodes.[11] In particular, the introduction of scalable switching must be at the center of scalable network growth.

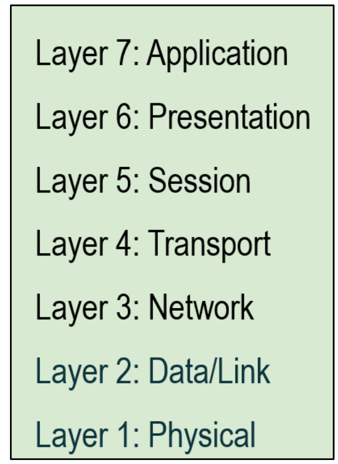

Figure 1: Networking layers

Adopting SDN principles for quantum networking provides a well-defined approach to scaling up their design and control.[12] SDN relies on a layered representation of the network functionality in which abstraction of communication supports the design of hierarchical protocols for programming the network behavior.[11,13] Figure 1 represents a typical layering of network protocols based on the abstraction of different concerns. Examples include how signals are transmitted between nodes, how switches establish quantum channel between nodes, and how separate quantum networks can be inter-networked.[14] The purpose of SDN is to manage the interactions between nodes by specifying policies for how different layers behave. In particular, we will use SDN to program the functionality of the switching that is defined by the link and physical layers. This approach will provide more convenient management of the network as well as well-defined notions of reusability, tunability, and networking scaling.

Each layer in the networking stack of Fig. 1 expresses a unique concern in establishing scalable communication. The lowest layer represents the physical encoding of signals and specifies how signals are transmitted along the wires, fibers, or free-space medium that exists between devices in the network. Commonly abbreviated as PHY, this layer is especially unique for quantum networks as it specifies how classical in addition to so-called quantum signals encode information.[11] Layer 2 is the data link layer and it defines how binary messages and quantum states are encoded into the classical and quantum physical signals, respectively. The link layer is also the lowest layer that defines an addressing mechanisms for network devices based on hardware addresses. In particular, a layer-2 switch uses these hardware or medium access control (MAC) addresses to bridge communications between devices within a local area network. The extension of addressing to include communication between other networks, i.e. inter-networking, is established in Layer 3. Layer 3 uses network addresses, for example IP addresses, to route communication across networks and to forward traffic to the next device along a route. In conventional networks, this traffic is defined by packets that encapsulate the data defined by high-lying layers in the networking stack. Layers 1 through 3 suffice for software-defined quantum network switching on a local area network. However, higher-order inter-network protocols are required for switching for long-range communication. For example, layer 4 is responsible for transport, which specifies how communications are partitioned into multiple packets and reconstituted by terminal devices.

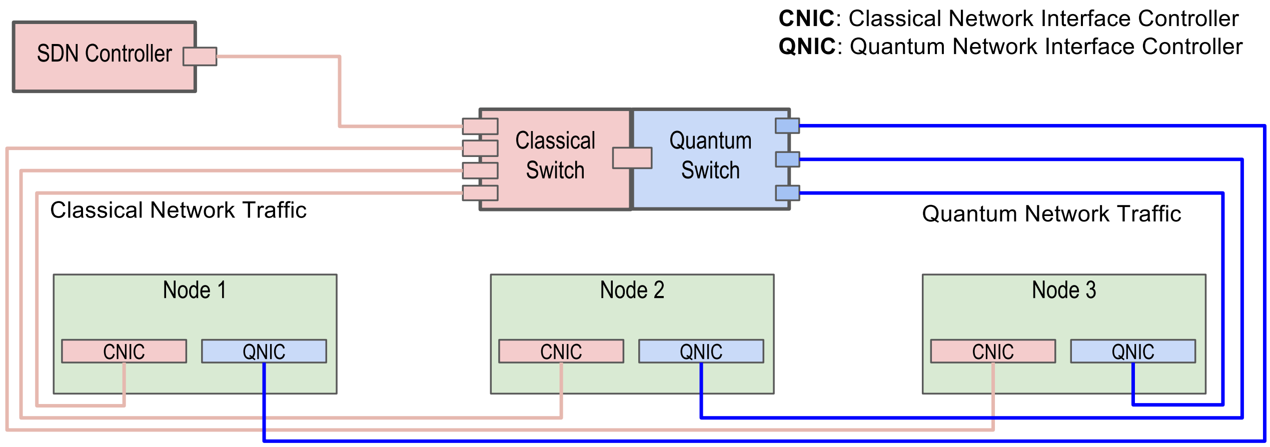

Figure 2: A schematic of a 3-node network including switch and network controller.

Our architecture for the quantum network consists of both classical and quantum layers. We adopt a model for the quantum network that mimics the layering shown in Fig. 1, but modified to account for uniquely quantum communication concerns. For example, the physical layer is assumed to consist of classical and quantum sub-layers, which embody the encoding and transmission of classical and quantum information, respectively. We further consider the link layer to embody concerns for linking together nearest-neighbor nodes in local area quantum networks (LAQN's).

SDN switches offer methods for managing network traffic that modify how packets arriving at the switch are forwarded. We use this model for the physical and link layers in a local-area quantum network to designate specific quantumand classical datapaths that define how information flows through the network nodes. In particular, we use SDN to manage how quantum datapaths are constructed within switches. This approach relies on separating the control and data planes within the switch. In order to provide a concrete realization of these ideas, we use the OpenFlow protocol to express the difference between the control plane and data plane for purposes of quantum network switching. The OpenFlow protocol specifies how an administrator reconfigures switches and routers within a network by defining the control plane. We adapt the OpenFlow protocol to management of a quantum network and specifically management of the quantum switch. We demonstrate the protocols and software needed to carryout switching within a 3-node quantum network.

3. QUANTUM NETWORK SWITCHING PROTOCOLS

An example of a 3-node network is given in Fig. 2. Each node in the network consists of a classical network interface controller and a quantum network interface controller, which provide access to the classical and quantum channels respectively. The opposing end points for these channels are a classical or quantum switch. These two switches are responsible for routing their respective signals between nodes. In addition, an external controller is present to reconfigure the classical switch. Given this design for the quantum network architecture, we create an explicit protocol for communicating quantum information between users within the switched network. Our protocol relies on classical communication between the nodes to establish synchronization of quantum transmissions. For example, when Alice wishes to communicate with Bob, she first establishes a classical connection through the switch. The classical switch queries the state of the attached quantum switch to determine if a quantum channel can be established between the source and destination. If the route is possible, the switch forwards Alice's request to Bob, who will acknowledge that he is ready to receive the transmission. If the route can not be established, the switch notifies Alice.

3.1 Switching in a Quantum Network

Central to this network design is a switch capable of performing existing classical networking functions, such as switching and routing. In addition, the classical switch is tightly coupled to the quantum switch and is responsible for its configuration. This classical switch should be knowledgeable about the capabilities of the quantum switch and respond accordingly to unsupported requests by network hosts. This necessitates knowledge of the quantum communication protocols that are to be used by the clients. For our implementation, we use a popular software-defined network switch called Open vSwitch (OVS),[15] an open-source and production quality virtual network switch. We adapt OVS to be coupled to a quantum switch, which represents the ability to route the optical paths of incoming photons from source to destination. The purpose of coupling between these two switches is to enforce the protocol for how quantum signals are transmitted between end points and to avoid collisions. In this protocol, when Alice wishes to transmit quantum data to Bob, she does so by sending a handshake packet with the data structure given in Fig. 4. These packets are exchanged during the protocol to synchronize transmission and reception of quantum communication by the source and destination. The protocol packets must contain the relevant metadata for the endpoints to either transmit or receive quantum data successfully. A sequence diagram outlining our switching protocol is visualized in Fig. 3. Alice serves as the source in this example and her node is composed of both a classical network interface card (CNIC) and a quantum network interface card (QNIC). The CNIC and QNIC are to transmit classical and quantum signals respectively. Similarly, Bob is the destination for this transmission and his node also consists of a CNIC and QNIC. A network switch is also decomposed into its quantum and classical elements. Alice and Bob are both connected to this switch and others may be connected as well. However, we do not include those other nodes in the sequence diagram.

In this first stage of the protocol, Alice's CNIC transmits a protocol packet that uses Bob's network address as the packet destination. This request packet is processed by the classical switch. The switch identifies the protocol packet as having the quantum switching structure and parses the protocol data. The classical switch then sends a configuration request to the quantum switch querying whether a quantum channel can be established between the source and destination. The quantum switch re-configures its underlying hardware to establish the desired quantum channel and returns a confirmation. In order to prevent another host from interrupting this quantum transmission, the classical switch will not reconfigure until Alice's CNIC issues another packet signalling the end of the quantum transmission, in effect locking the state of the quantum switch. Additional details on the implementation of the classical switch processing with respect to OVS can be found in the subsequent section. After processing on the protocol packet is complete, the classical switch forwards the handshake packet to Bob's CNIC. After Bob parses the request packet, he transmits a reply packet to Alice via the classical network. Bob's reply packet, which uses the same data structure, is also by the classical switch. However, the reply packet is forwarded through a different path than Alice's packet. In this processing path, the switch understands that the packet is in response to Alice's request and the reply packet is then forwarded from the switch to Alice.

In the second stage of the protocol, Alice's QNIC transmits quantum signals to the quantum switch. The quantum switch, having already reconfigured to provide a physical link between the ingress and egress quantum ports, is effectively a pass-through device, allowing Alice to transmit to Bob. More advanced networks may provide local quantum state buffering at the switch or implement a repeater pattern to increase reliability of the transmitted state. After Alice has transmitted her quantum data, she closes the quantum channel by transmitting a classical packet to Bob. This final stage of the protocol mimics the opening handshake. The second protocol packet from Alice's CNIC triggers the classical switch to release the lock on the quantum switch, allowing other hosts to now utilize it and transmit over the quantum channel. Bob acknowledges this closure by replying with an acknowledgement packet.

Our first generation protocol is designed to give much flexibility and provide opportunity for refinement for specific use cases. Our protocol packet structure is visually detailed in Fig. 4. There are three main fields which help facilitate negotiations between hosts for quantum transmissions. The first field is a protocol identifier which is used to discriminate between transmissions such as direct quantum transmission, super-dense coding, quantum teleportion, and various permutations on these concepts. The action field is used to discriminate between packets used to initialize the quantum transmission session and packets used to terminate the transmission session. The last field is a collection of flags which are used to experiment with ideas such as priority transmission, whether the client should expect another initialization packet soon. These are the three fields which allow for basic functionality and it also meets the requirement that metadata can be extracted by the classical switch to configure the quantum switch. It should be expected that future work in this field will certainly modify this structure and specialize the packet for various use cases. The packet structure for the next generation protocol will include information such as error correction techniques used, the maximum transmission rate, and information on frequency or time division multiplexing.

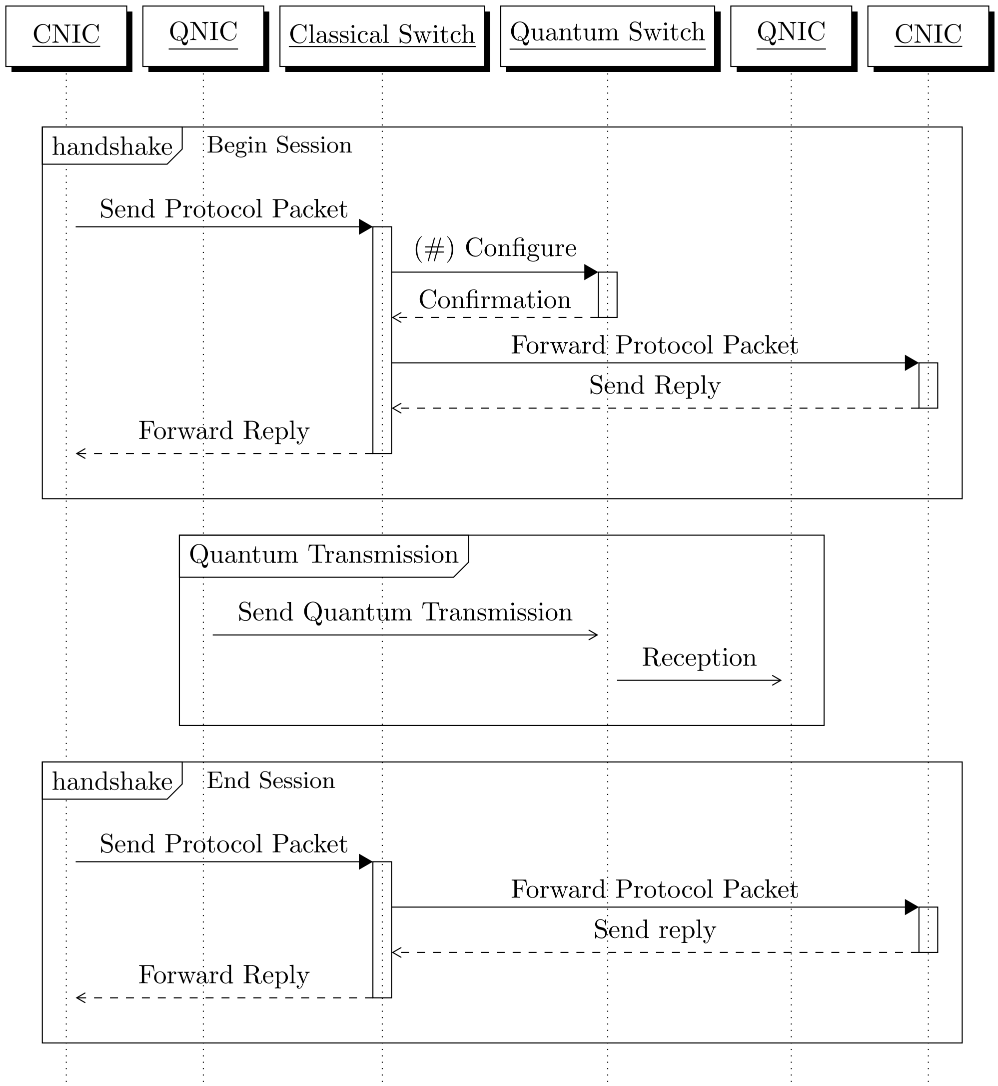

Figure 3: The linking protocol for transmission of quantum information between two hosts where the left hand side represents the transmitting node and the right hand side represents the receiving node. The classical (CNIC) and quantum (QNIC) network interfaces relay the corresponding traffic on the network.

3.2 OpenFlow Controller and Customized Actions

Management of the network is performed by an external controller,[16] which provides a high-level programming paradigm for network behavior. The controller uses the OpenFlow protocol to instruct switch behavior for incoming packets. We utilize the flexibility of the OpenFlow protocol to design new switch actions. In particular, we have implemented a new action in the OVS switch that configures a quantum switch. The principle of SDN is that the network controller configures the switch. Therefore, we use the OpenFlow protocol to configure OVS such that whenever it encounters the specific quantum handshake, it configures the quantum switch which responds accordingly to a set of prescribed rules. Unlike conventional networking protocols, the classical switch reconfigures the quantum switch to support the necessary route by extracting relevant properties from the handshake packet. We have designed our implementation in OVS to separate the state of the classical transmission from the state of the quantum channel. This reduces the complexity of the quantum switch and leverages the existing capabilities of OVS to translate information between the networking layers.

In practice, the switch manages multiple actions to process the incoming classical packets. Our action

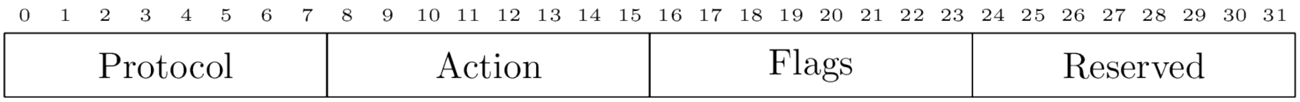

Figure 4: The packet structure for our first generation handshaking protocol.

metadata is transmitted in the form of packets that have a header consisting of an action type, the packet length, and an experimenter ID.[17] The body of the OpenFlow packet specifies the parameters used by the classical switch to configure the quantum switch in response to specific conditions 5. We use two fields for our experimenter actions that specify a quantum communication protocol version and the protocol options. Two additional fields in the specification provide the physical input and output ports for the quantum switch. The controller issues guidance on how the switch handles interaction between every pair of users to which it connects. Once the switch learns this behavior, it operates independently from the controller. Consequently, our basic handshaking protocol between users can accommodate a variety of quantum communication methods, including super-dense coding and quantum teleportation.

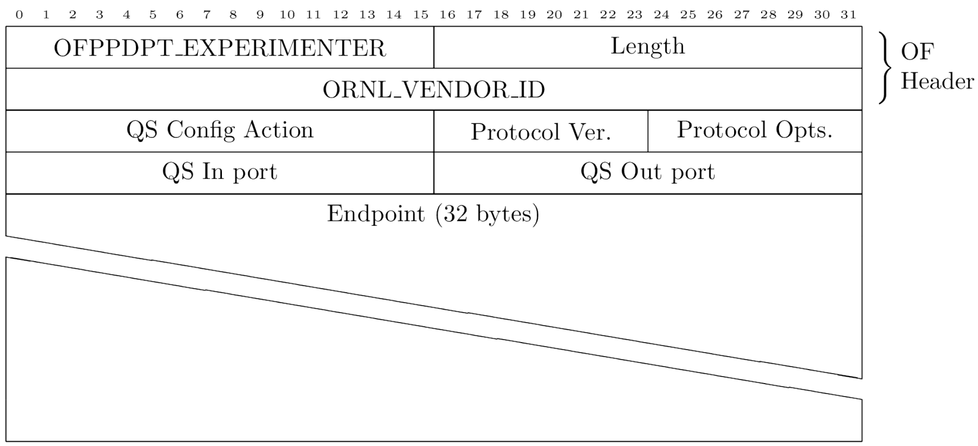

Figure 5: The OpenFlow action packet the controller transmits to the classical switch to construct a quantum protocol aware flow. The metadata within the packet is used in configuration of the quantum switch referenced by the endpoint field.

When the action is stored within OVS, an additional metadata field is included. This field contains a connection pointer to a shared socket used to connect the classical switch to the quantum switch. This is achieved through an abstraction layer that allows multiple actions to use the same endpoint, thereby reducing the number of socket objects required. The usual thread protection mechanisms are implemented in this layer making scaling to multiple threads possible.

This new action is added to a flow to match on quantum handshake packets. When a handshake packet is receiving, it is first determined if the packet constitutes an opening or closing of a quantum transmission. If it is an opening, the classical switch using the metadata stored with the action to contact the quantum switch to configure a quantum link to be established between the ports belonging to the requester and requestee. The quantum switch will return whether or not the configuration is successful and the classical switch will then set a lock on the quantum switch which is unlocked when a closing packet is received. This prevents other users from interrupting the ongoing quantum transmission and allocates resources in a first-come, first-served pattern.

Additionally the ingress and egress quantum ports may be cached within the classical switch for performance purposes on subsequent transmissions. This reduces the overhead associated with configuring the quantum switch if one client sends a burst of quantum transmissions to the same receiver. This scheme works only if a guarantee is made that the state of the quantum switch can only be modified by the classical switch. More complex caching schemes may be used, for example, preferring one of several simultaneous requests caught by the classical switch. This would reduce the number of state transitions the quantum switch would make in a given length of time.

4. CONCLUSION

We have presented a new concrete methods for quantum network switching based on software-defined networking principles. Our approach builds on the modern use of SDN switches to add new actions to the flow tables responsible for classical packet forwarding. By separating the classical and quantum data planes, we have used the classical switch data path to instantiate virtual quantum circuit between quantum end points. We have provided a switching protocol based on classical handshake packets between endpoints to establish synchronization of the quantum transmission and reception as well as to build up and tear down the virtual quantum circuits. We have further provided a specification for the payload exchanged during the handshake with an emphasis on the metadata required to perform the subsequent quantum communication.

In addition to our specification of the switching protocol, we have presented an implementation of these ideas using open vSwitch and the OpenFlow protocol. We have leveraged prior development of the application layer for driving a super-dense coding experiment and a numerical simulator for tracking the quantum channel preparation, transmission, and detection. Our simulation results validate the protocol for a small, three-host network. Moreover, this quantum networking software is immediately ready to be extended in experimental networking environments. We have begun to realize communication between nodes using super-dense coding in the laboratory.[18] These results will lead to an understanding of the challenges facing the engineers of future quantum networks.

REFERENCES

[1] C. Elliott, "Building the quantum network," *New Journal of Physics*, vol. 4, no. 1, p. 46, 2002.

[2] T. Lnger and G. Lenhart, "Standardization of quantum key distribution and the etsi standardization initiative isg-qkd," *New Journal of Physics*, vol. 11, no. 5, p. 055051, 2009.

[3] T. Scheidl, R. Ursin, A. Fedrizzi, S. Ramelow, X.-S. Ma, T. Herbst, R. Prevedel, L. Ratschbacher, J. Kofler, T. Jennewein, and A. Zeilinger, "Feasibility of 300km quantum key distribution with entangled states," *New Journal of Physics*, vol. 11, no. 8, p. 085002, 2009.

[4] T. E. Chapuran, P. Toliver, N. A. Peters, J. Jackel, M. S. Goodman, R. J. Runser, S. R. McNown, N. Dallmann, R. J. Hughes, K. P. McCabe, J. E. Nordholt, C. G. Peterson, K. T. Tyagi, L. Mercer, and H. Dardy, "Optical networking for quantum key distribution and quantum communications," *New Journal of Physics*, vol. 11, no. 10, p. 105001, 2009.

[5] T.-Y. Chen, J. Wang, H. Liang, W.-Y. Liu, Y. Liu, X. Jiang, Y. Wang, X. Wan, W.-Q. Cai, L. Ju, L.-K. Chen, L.-J. Wang, Y. Gao, K. Chen, C.-Z. Peng, Z.-B. Chen, and J.-W. Pan, "Metropolitan all-pass and inter-city quantum communication network," *Opt. Express*, vol. 18, pp. 27217–27225, Dec 2010.

[6] W. Weigel and G. Lenhart, "Standardization of quantum key distribution in etsi," *Wirel. Pers. Commun.*, vol. 58, no. 1, pp. 145–157, 2011.

[7] D. Elkouss, J. Martinez-Mateo, A. Ciurana, and V. Martin, "Secure optical networks based on quantum key distribution and weakly trusted repeaters," *J. Opt. Commun. Netw.*, vol. 5, pp. 316–328, Apr 2013.

[8] D. Wu, W. Yu, B. Zhao, and C. Wu, "Quantum key distribution in large scale quantum network assisted by classical routing information," *International Journal of Theoretical Physics*, vol. 53, no. 10, pp. 3503–3511, 2014.

[9] A. Ciurana, V. Martin, J. Martinez-Mateo, B. Schrenk, M. Peev, and A. Poppe, "Entanglement distribution in optical networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, pp. 37–48, May 2015.

[10] H. Barnum, C. M. Caves, C. A. Fuchs, R. Jozsa, and B. Schumacher, "Noncommuting mixed states cannot be broadcast," *Phys. Rev. Lett.*, vol. 76, pp. 2818–2821, Apr 1996.

[11] T. S. Humble, "Quantum security for the physical layer," *Communications Magazine, IEEE*, vol. 51, no. 8, pp. 56–62, 2013.

[12] V. R. Dasari, R. J. Sadlier, R. Prout, B. P. Williams, and T. S. Humble, "Programmable multi-node quantum network design and simulation," *Proc. SPIE*, vol. 9873, p. 98730B, 2016.

[13] T. S. Humble and R. J. Sadlier, "Software-defined quantum communication systems," *Optical Engineering*, vol. 53, no. 8, p. 086103, 2014.

[14] R. J. Sadlier and T. S. Humble, "Superdense coding interleaved with forward error correction," *Quantum Measurements and Quantum Metrology*, vol. 3, pp. 1–8, 2016.

[15] www.openvswitch.org, accessed October 2016.

[16] V. R. Dasari and T. S. Humble, "OpenFlow arbitrated programmable network channels for managing quantum metadata," *Journal of Defense Modeling and Simulation*, vol. 14, p. 1, 2016.

[17] www.opennetworking.org, accessed October 2016.

[18] B. P. Williams, R. J. Sadlier, and T. S. Humble, "Superdense coding over optical fiber links with complete bell-state measurements," *Physical review letters*, vol. 118, no. 5, p. 050501, 2017.