Quantum Internet Alliance M4.2: List of atomic tasks

H. Ollivier

2020-02-24

1 Purpose

Application level protocols need to have access to networking services such as entanglement sharing between any two points of the network. While such service is at the heart of the quantum internet architecture, additional functionalities can be required or just convenient to have for better, faster, wider development of application level protocols.

The purpose of this report is to review a wide range of such protocols searching for atomic repeatable functions while categorising them along several dimensions. By doing so, we aim at providing building blocks that:

- lessen the amount of code and control needed while developing applications for the quantum internet (in particular through code reuse);
- allow benchmarking of the nodes and network capabilities against these tasks;
- provide functionalities with sound cryptographic definitions;
- provide a simulation platform where these functions would already be implemented, to further accelerate the creation cycle of quantum protocols.

2 Methodology

1. Review of the quantum protocol zoo looking at the various protocols;
2. Identify and group candidate atomic functions;

3. Categorising various candidates into network stages, type (quantum internet layer attribution, off-layer), and necessity

4. Integration into protocol zoo's knowledge graph

3 Review of the quantum protocol zoo

Protocol	Functionalities u
GHZ-based Quantum Anonymous Transmission https://arxiv.org/abs/quant-ph/0409201	Classical authent GHZ creation an Classical collisior Single qubit mea Single qubit Had Limited memory Teleportation
Verifiable Quantum Anonymous Transmission https://arxiv.org/pdf/1811.04729.pdf	Notification (priv Single qubit mea Imperfect GHZ s Limited memory (Uses GHZ anon
Polynomial Code based Quantum Authentication https://arxiv.org/pdf/quant-ph/0205128.pdf	Clifford circuits (Memory
Fast Quantum Byzantine Agreement https://dl.acm.org/doi/10.1145/1060590.1060662	Distribution of G Verification of n- (Uses oblivious c (Uses verifiable G
Quantum Bit Commitment https://arxiv.org/abs/1108.2879	BB84 encoding o Single qubit mea Secure classical c Fast operations t
Quantum Coin Flipping https://arxiv.org/abs/quant-ph/9904078	$\pi/9$ single qubit Multi qubit POV
Gottesman and Chuang Quantum Digital Signature https://arxiv.org/abs/quant-ph/0105032	Memory Swap test Stabilizer states
Prepare and Measure Quantum Digital Signature (QDS) https://arxiv.org/abs/1403.5551	BB84 encoding BB84 decoding
Measurement Device Independent QDS https://arxiv.org/pdf/1704.07178.pdf	Classical authent Measurement De BB8484 Encodin
Multipartite Entanglement Verification https://www.nature.com/articles/ncomms13251	Authenticated cl Secure classical b Common shared Limited memory BB84 Measureme GHZ source / br
Quantum Fingerprinting https://arxiv.org/abs/quant-ph/0102001	Clifford gates Swap test
BB84 https://core.ac.uk/download/pdf/82447194.pdf	BB84 Encoding a Authenticated cl Privacy amplifica Information reco
Device Independent QKD https://arxiv.org/abs/1811.07983	EPR distribution Information reco
Quantum Leader Election	(Uses Weak coin

4 Task extraction and categorisation

The table below presents the extracted tasks from the protocols listed above. Each task is categorized in the layer model of quantum networks (ie. Physical, Layer, Network, Transport, Session, Presentation, Application or Off layer). The Network stage is also specified for each of these atomic functions (ie. in increasing complexity starting from trusted repeaters, entanglement distribution, quantum memory and quantum computing).

Function	Layer	Network st
Sending qubit	Transport	Trusted rep
Sending qubit blocks	Transport	Trusted rep
Teleportation protocol	Transport	Entanglem
Creation and braodcast of GHZ state	Session	Quantum r
Creation and broadcast of any stabilizer state	Session	Quantum r
Creation and broadcast of arbitrary graph states	Session	Quantum c
Quantum One Time Pad / confidential channel (encoding and decoding)	Presentation	Trusted rep
BB84 Encoding of classical data	Presentation	Trusted rep
BB84 Decoding to classical data	Presentation	Trusted rep
Single Qubit Preparation in equatorial plane (finite set of angles)	Presentation	Trusted rep
Single Qubit Measurement in equatorial plane (finite set of angles)	Presentation	Trusted rep
Multi qubit POVM	Presentation	Quantum r
Local Pauli gates	Off	Quantum r
Local Clifford gates	Off	Quantum r
Local memory	Off	Quantum r
Non Clifford gates	Off	Quantum c
Verification of stabilizer state	Off	Prep. & m
QFactory	Off	Quantum c
Swap Test	Off	Quantum r
Information reconciliation	Off	Classical
Classical error correction	Off	Classical
Privacy amplification	Off	Classical
Secure classical broadcast channel	Off	Classical
Classical authenticated channel	Off	Classical
Quantum 1 way function	Off	Prep. & m
Anonymous transmission channel	Session	Quantum r
Quantum Authenticated Channel	Session	Quantum r
Weak String Erasure	Application	Trusted rep

5 Tasks specifications

5.1 Sending qubit

This functionality should be provided at the transport layer for entanglement sharing networks. It would be at the link layer for networks based on direct transmission of quantum informaton (QKD-like networks).

For a transport layer functionality, it is expected to provide reliable service with flow control (allowing the sender to improve its probability of correct transmission by detecting possible congestion at the receiving end or along the way).

Inputs	Outputs
Source node: Qubit	Source node: ACK and Flow Control - Congestion Control Message /
Source node: Target node ID	Target node: Qubit
Source node: Metadata	Target node: Qubit ID, Source node ID, Additional Qubit metadata

5.2 Sending qubit blocks

Same as previous functionality but handling several qubits in a single call.

Inputs	Outputs
Source node: Block of qubits	Source node: ACK and Flow Control - Congestion Control
Source node: Target node ID	Target node: Block of qubits
Source node: qubit and block metadata	Target node: Block of qubit IDs, Source node ID, Addition

5.3 Teleportation halves

While teleportation is the heart of the transport layer in entanglement sharing networks, it is useful to have each half of the protocol available directly to the programmer as a way to teleport while quantum one time padding information. This is especially useful in measurement based quantum computation for blindness and verification. In this case, teleportation measurement is performed, but the sender does not send the corrections to the receiver.

5.3.1 Sending half

Inputs	Outputs
Source node: Qubit or block of Qubits	Source node: Corrections or block of corrections
	Target node: Uncorrected qubit or block of uncorrected qub

Correction sending and metadata sending is left up to the Source node as protocols might require a full flexibility over what is sent (or not sent).

5.3.2 Receiving half

Inputs	Outputs
Target node: Corrections or block of corrections	Target node: Corrected Qubit or block of Qubits

The receiving half can be replaced by a quantum One-Time-Pad encoding / decoding with a proper labelling of the corrections.

5.4 Creation and braodcast of GHZ state

GHZ states are central to several multi-party protocols. Several implementations can be proposed:

- either a centralized create and broadcast operation using a regular transport layer,
- or tapping directly into the network layer for using bipartite entanglement as a way to generate the GHZ state in a decentralized fashion.

The first option would yield two atomic sub-tasks, while the latter would involve a protocol on its own.

5.4.1 Local prepare

Inputs	Outputs
Number of parties	N-Party GHZ state

5.4.2 Broadcast

Inputs	Outputs
Source node: N-qubit register	Source node: ACK and Flow Control - Congestion
Source node: length-N Target node IDs	Per Target node : Qubit
Source node: Additional constraints (eg. TTL)	Per Target node: Qubit ID, Source node ID, metad
Source node: Overall state metadata	

5.5 Creation and broadcast of any stabilizer state

Specifications similar to GHZ state creation and broadcast.

5.6 Creation and broadcast of arbitrary graph states

Specifications similar to GHZ state creation and broadcast.

5.7 Quantum One Time Pad / confidential channel (encoding and decoding)

Encoding and decoding are identical, yielding a single atomic task that is entirely performed locally. The "send" part can be performed using the send qubit atomic function and would yield a confidential quantum channel. It can also be implemented directly via teleportation without revealing the corrections.

Inputs	Outputs
Block of qubits	Block of encoded qubits
Block of encoding key (ie 2 bits per qubit specifying X and Z rotations)	

5.8 BB84 Encoding of classical data

One of the most useful encoding, it is used in many protocols and is also helpful in unit-testing the library itself. This would be naturally embedded in a slightly more general preparation functionality where all 6 states that are eigenstates of Pauli operators can be prepared at will.

Inputs	Outputs
Block of classical data bits	Prepared block of qubits
Block of encoding bases	

5.9 BB84 Decoding to classical data

Same reason. Symmetric functionality.

Inputs	Outputs
Block of qubits	Decoded block of classical bits (measurement outcomes given the measuren
Block of encoding bases	

5.10 Single Qubit Preparation in equatorial plane (finite set of angles)

Useful in the context of measurement based quantum computing. A natural set of preparation angles is $\{\frac{k\pi}{4}\}_{0\leq k\leq 7}$ as it is widely used in measurement

based quantum computing schemes. Other angles could be optionally considered.

Inputs	Outputs
Block integers specifying the preparation angle	Block of prepared qubits
Block of bits (to apply $+\pi$ to the preparation)	
Option: list of preparation angles, defaults to $k\pi/4$	

5.11 Single Qubit Measurement in equatorial plane (finite set of angles)

Same reason. Symmetric functionality.

Inputs	Outputs
Block integers specifying the measurement angles	Block of classical bits (measurement outcomes
Option: list of measurement angles, defaults to $k\pi/4$	

5.12 Multi qubit POVM

While it is not possible to allow the implementation of generic POVM's even on a few qubits, some should be available readily as atomic functions. A possibility would be to allow for POVM's obtained as observables defined by stabilizer measurements as it would allow completing basic quantum error correcting schemes.

Inputs	Outputs
N-qubit state	bit (measurement outcome associated to the projectors onto the ± 1 eig
Length-N Pauli operator P	

5.13 Local Pauli gates

Necessary. Mostly provided by the backend it self.

Inputs	Outputs
Qubit	Rotated qubit
Pauli operator	

5.14 Local Clifford gates

Same as previous functionality.

5.15 Non Clifford gates

Most backends or hardware provide the ability to perform T gates. These will serve as building blocks for distillation schemes before being used in real circuits. Same specification as the previous functionality.

5.16 Local memory manager

The idea behind this functionality is to be able to handle various scenario influencing the robustness of protocols, such as being able to tell neighboring nodes that the capacity to accept new qubits is low before they actually send information. It might also be interesting to be able to keep track of the time to live of various qubits so that other parts of the protocols can take that into account and give higher priority to operations with a lower TTL.

The implementation of such functionality can be done in various ways and will highly depend on the backend and chosen architecture.

Inputs	Outputs
nil	Available capacity for reciving or creating new qubits
	Optionally active qubit ID and their remaining TTL

5.17 Multi-site verification of stabilizer state

This allows verified multiparty protocols. It also enters into unit testing for multi-party operations.

5.17.1 Verifier side

Inputs	Outputs
Length-N Prover node ID's	GOK / NOK
Qubit ID's per prover node	Accepted qubit-ID per Prover node
Stabilizer state description	

5.17.2 Prover side

Inputs	Outputs
Qubit IDs	list of measured outcomes
List of measurements to perform	

5.18 QFactory

Blind remote preparation of a classical described quantum state.

Inputs	Outputs
Client node: classical description of the quantum state	Server node: quantum state Client: classical bit spceifying the computat

5.19 Swap Test

Implies to add the swap gate first and then the test. It should work indistinctively for individual qubits and blocks of qubits. The returned classical bit corresponds to a single measurement outcome (ie the test needs to be repeated to get the overlap value between the two tested-states).

Inputs	Outputs
2 blocks of qubits	classical bit

5.20 Information reconciliation

Useful for implementations of QKD like protocols. Would offer a template for reconciliation allowing to plugin various error correction schemes.

Inputs	Outputs	Parameters
Each participating node: Sifted key bits	Each participating node: reconciled key bits	Scheme to u Error estima

5.21 Classical error correction

Many protocols require classical error correction at a very fine grained level (be it for exploiting classical code properties in algorithms or to have manual control over some quantum error correction schemes using CSS codes). This entry would be consisting of a sub-library providing encoding, error estimation and decoding procedures as well as code manipulation functionalities. Depending on the use of these codes (either as being used off-line or on-line) they should provide fast implementations (possibly requiring dedicated libraries or even dedicated hardware).

5.21.1 Classical information encoding

Inputs	Outputs
(n,k)-code Length-k classical bits	Length-n encoded block

5.21.2 Classical information error estimation and decoding

Inputs	Outputs
(n,k)-code	Length-k decoded bits or quanti
Length-n noisy encoded block (bits or quantized continuous value)	Length-$(n-k)$ syndrome values
	Most likely length-n error vecto

5.21.3 Classical codes manipulation

Series of functions such as:

- Coset sampling
- Dual code encoding and decoding
- Concatenation
- ...

These functions might be provided by specialized libraries.

5.22 Privacy amplification

Useful for implementations of QKD protocols. Would offer various schemes for privacy amplification

Inputs	Outputs	Parameters / Shared resources
Insecure shared random key	Secure shared random key	Privacy amplification scheme
		Estimate of the information in the hands

5.23 Quantum 1-way function

Used in digital signature schemes and quantum cheques. This function takes a classical bit, a key and outputs a block of qubits that encode the classical bit with the help of a classical key k.

The inputs/outputs is defined in a way similar to the BB84 encoding / equatorial plane encoding

5.24 Channels

Various implementation of quantum and classical channels will be useful as it is common in protocols to require one or several of these. Their precise specification will be depending on the network available to implement them. One constraint needs to be kept in mind: most implementations will be required to have low latency as, even for the case of classical information, it might otherwise mean that qubits containing precious quantum information are at risk of decoherence.

The Flow control and Congestion control messages available for sending qubits and blocks of qubits will need to be adapted to theses channels.

List of channels to implement:

- Secure classical broadcast channel
- Specific implementation could be required for high precision timing reasons.
- Classical authenticated channel
- Quantum anonymous transmission channel
- Quantum Authenticated Channel

5.25 Weak String Erasure

Protocol per se, but used as a component in other protocols. Implementation shall rely on BB84 preparation and decoding.

Inputs	Outputs
Source node: length-N bit string of information to send	Source node: ACK and Flow Control - Con
Source node: length-N bit string of basis choices	Target node: Set of bits and positions wher
Target node: length-N bit string of basis choices	