Design and Implementation of Network stack and Protocols for a Multipartite entanglement-based Quantum Network

Arahant Ashok Kumar (aak700)

Abstract

Herein I explore some of the shortcomings of classical network in addressing critical security problems that plague the internet, and elaborate on the advantages of the quantum approach, thus presenting my case for the design and development of a quantum network based on multipartite entanglement.

The evolution of the classical internet, although quite patchy, is extra-ordinary. It is the fastest penetrating infrastructure today, eclipsing electricity, telephone, gas, etc. Internet has fundamentally revolutionised the way we access information about this world and communicate, a majority of which happens over the internet.

However, there are many critical problems associated with classical internet, such as, Spam (SPIT), Security (DoS, TCP SYN flooding, Data breach attacks), and Remote voting (by mail or online), where classical network and communication protocols fall short. Quantum computing and its communication protocols are well suited to provide elegant solutions to these problems.

Finally, I've designed a quantum network framework and backed it with an elaborate implementation and simulation and used data analytics on the simulation data. I've used Qutech's SimulaQron for this purpose. It provides a fairly open and programmable framework to implement, which abstracts the nature of the underlying quantum technology.

Introduction

I shall start by introducing some of the problems with **Classical network technology:**

1. **SPIT** - VoIP spam (issues with Authenticity and Privacy)
2. **Privacy** is one of the biggest concerns with exponential digitisation
3. **Denial of Service** is the 2nd most popular [7] vulnerability
    1. **TCP SYN flooding** - DoS attack on servers
4. **Data breach** is the 5th most popular [7] attack. While this is still damaging at the corporate level, it has serious consequences in govt institutions.
5. **Vote by mail**: With the Covid-19 pandemic, in-person voting is become a fatal choice and vote by mail has its own set of problems

Quantum Computing - potential use cases

Below are some practical applications of quantum computing protocols.

1. Central bank Quantum encryption

Critical govt institutions such as Central banks, Intra-governmental communication, Election commissions, Intelligence agencies etc. who have a lot to protect and to lose. A few years ago, the central bank of Bangladesh was hacked.

2. Quantum Voting

I recently came across an article [8], which talks about turning to vote-by-mail. The major problem here is verifying authenticity of vote - voter fraud [9], ballot harvesting [10] - integrity and confidentiality of the ballot envelopes, privacy of voters which can lead to serious legal disputes [11].

Quantum communication protocols can provide elegant solutions to this complex classical problem. Given that qubits cannot be tampered with or cloned, they could be the the perfect solution. The votes in the form of classical bits can be encoded into qubits, and transmitted through quantum communication protocol(s). All this is possible by protecting citizens' privacy, effectively making democracy more reachable.

3. End-to-end Quantum encryption

Quantum Key Distribution can be used to provide end-to-end encryption, with a high probability of success.

4. Mobile [proxy] Quantum Computing

Although it isn't feasible to build a hand-held or even a mobile quantum computing device, yet, it can be used to connect to quantum servers, through QKDs. QKD provides a very secure way of generating and sharing secret keys between two entities, and once generated, it can be used to encrypt in the usual manner.

Quantum Computing concepts

Qubit

Qubit is the quantum equivalent of a classical bit, although completely different in nature. It stores one unit of quantum information, in terms of states. Information is stored as a linear combination of orthogonal bases. Practically, it might be a photon, which can be a linear combination of vertical ($|1\rangle$) and horizontal ($|0\rangle$) polarisation. A single qubit, $|Q\rangle$, is most commonly represented in the $\{|0\rangle,|1\rangle\}$ and $\{|+\rangle,|-\rangle\}$ bases. It can also be electrons with spin polarisation.

Superposition and Measurement

A qubit can represent multiple values simultaneously. A crude analogy would be the dual nature of light, where it exists both as a particle and a wave simultaneously, and reveals its nature depending the experiment and expectation. Similarly, a qubit is a superposition (linear combination) of multiple orthogonal states and it collapses to either one of the bases depending on the relative probability (amplitudes) of the bases and the measurement.

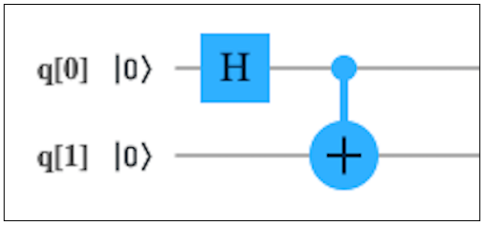
The uncertainty principle restricts anyone from tampering with the quiet without a high probability of losing the information it holds. This property of a qubit makes it well suited for many classical problems. Essentially, any tampering of the qubit is nothing but performing a measurement on it, which causes it to collapse, thus losing the information within with a high probability.

Entanglement

Mathematically, 2 qubits are entangled if their entangled state cannot be expressed as a tensor product of the 2 individual state. Measurement outcomes of the two qubits in any of the Bell states in the standard bases are either perfectly correlated or perfectly anti-correlated. The 4 bell pairs are: (1/sqrt(2))(|00〉+|11〉), (1/sqrt(2))(|00〉-|11〉), (1/sqrt(2))(|01〉+|10〉) and (1/sqrt(2))(|01〉-|10〉). Measuring the former two gives either |00〉 or |11〉 (perfectly correlated) and measuring the latter two leads to |01〉 or |10〉 (perfectly anti-correlated).

Entanglement involving 2 qubits are bipartite (Bell pairs) and those with more than 2 qubits are multipartite (GHZ, W). In the presence of bit-flip noise, multipartite states allow correction of error. This way, it is possible to improve fidelity.

Importantly, one Bell state can be converted into another through local unitary transformations (X and Z operators) and classical communication (LOCC). Entanglement is the bedrock of many quantum communication protocols such as CHSH, superdense coding, teleportation and swapping.



An Entangled Bell pair

Fidelity

In any real implementation of a quantum network, the generated entangled states will always differ from the perfect Bell states due to noise in the system. Fidelity measures how close a realised entangled state is to the ideal state. The Bell state to be useful has to have a minimum fidelity value.

Fidelity of a single instance of a quantum state cannot be measured, but can be estimated by measuring the quit error rate (QBER) of many generated states in succession. For a fixed basis, X, the $QBER_X$ is the probability of receiving equal measurement outcomes when measuring these qubits in this basis. Essentially, this is a probability distribution of the qubits over one of the orthogonal bases.

Purification

Purification raises the fidelity of an entangled state, essentially performing error correction, taking advantage of the specially-prepared initial state of the qubits. Purification takes two Bell pairs and attempts, via LOCC, into a higher fidelity state. This is a resource and time expensive process. [3] describes a scheduling algorithm to generate an entangled state with the desired min fidelity.

Heralded entanglement

As every entangled state doesn't have a minimum fidelity value, they need to be purified in order to increase their fidelity. This is resource and time expensive. In [1], they've devised a method to signal the participating nodes when an entanglement state (channel) with a minimum specified fidelity has been successfully created. This heralding signal is then transmitted to the nodes. Such signalling allows long-distance quantum communication without exponential overheads. This also helps in bookkeeping of the participating nodes in an entangled link and the state of the link itself.

Decoherence

Another important fact about qubits generated today is that they cannot be stored indefinitely and decohere very quickly. Quantum memories, which is a storage of qubits, are inherently noisy. This noise encountered by a qubit increases with the duration of storage. As a qubit encounters more noise its fidelity decreases.

Noise is also encountered in the channel. As the distance covered by a qubit increases its fidelity decreases. This practically limits the storage for long durations and transmission of qubits over long distances.

The Quantum advantage - Quantum Communication protocols

Various quantum communication protocols have demonstrated the quantum advantage

CHSH experiment

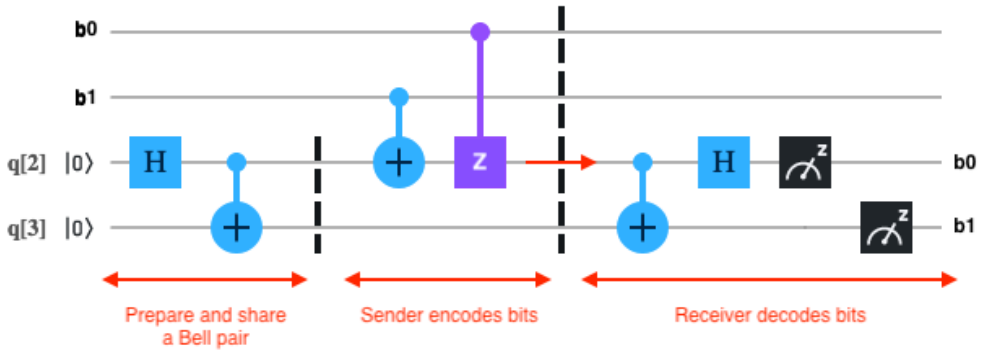
Suppose an entity one wishes to "transfer" information to another without communication, this experiment demonstrates how to achieve this with a maximum accuracy of ~85% using a quantum strategy, as opposed to the 75% through the classical strategy. The only requirement is that they share a Bell pair. Based on the outcome of a randomised experiment, the 2 entities decide the bases to measure their entangled qubits in. This essentially results in a maximum accuracy of $\cos^2(\text{pi}/8)$ (~85%). This demonstrates the quantum advantage over classical communication.

Superdense coding

This is a quantum communication mechanism through which 2 bits of information can be transferred through 1 qubit, with a prior entangled shared state between a sender and a receiver. The sender transforms her entangled qubit to one of the four bell states

through unitary operations (Z, X depending on the 2 bits), and sends it to the receiver, who then performs a CNOT operation with the sender qubit as the controller and the receiver's entangled qubit as the target, followed by a Hadamard operation on the control qubit. Measuring them extracts the 2 bits.

This reduces the size of information that needs to be transmitted and entanglement establishes a communication channel which ensures authenticity, integrity and confidentiality. However, it is a theoretical curiosity at this point given technology challenges.



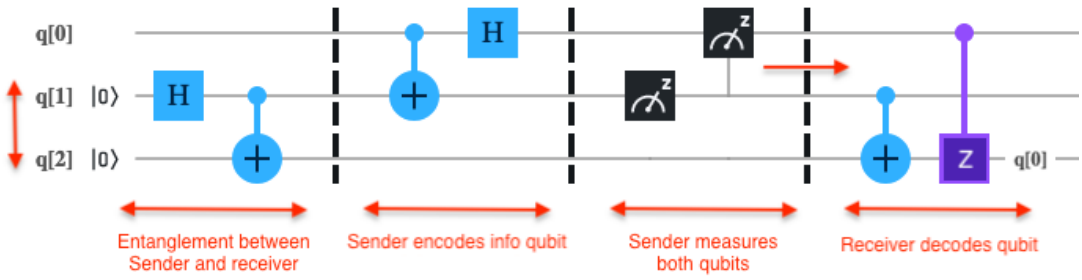
Through a shared Bell state, 2 bits can be packed into 1 qubit.

Quantum teleportation

Teleportation is a mechanism which makes use of entanglement between two participating entities to teleport qubits between them. Because of the no-cloning principle, the teleported qubit is destroyed at the sender and recreated at the receiver. Superdense coding requires a qubit to be transferred, which is a technological challenge and while CHSH doesn't require any communication, it doesn't guarantee 100% accuracy. Teleportation involves transferring classical bits over an already established entangled channel.

An entangled state is first shared between two entities. The qubit to be teleported is entangled with sender's half of the bell pair, and both these are measured. This results in a classical bit and destroys the quantum state of the qubit. When these two bits are sent to the receiver (2nd entity), they use them to perform LOCC on their half of the bell pair, eventually recreating the qubit from the sender (1st entity).

Here the random sequence of bits generated after measurement only makes sense when they're used to operate on the other entangled qubit. The advantage of teleportation over previous two is that here, information transfer can happen through classical bits over existing classical channels, while still ensuring 100% accuracy and the security that accompanies a quantum protocol. The classical bits are still vulnerable, but are practically useless without the corresponding entangled state used to encode them.



Quantum Teleportation in Circuit

Entanglement swapping

This is an extension of quantum teleportation. In the previous case, if the receiver is entangled with a 3rd entity, then the previously recreated (or teleported) qubit from the 1st entity can be further teleported to the 3rd entity by following the same steps, effectively, teleporting the qubit from the 1st to the 3rd entity, without being direct entangled.

No-cloning principle

Quantum information (qubit) cannot be copied, a restriction known as no-cloning theorem[4]. We cannot make multiple redundant independent copies of qubits, to store or retransmit later. The operator that comes close is CNOT. However, it fails when the input qubit is $|+\rangle$ which then outputs a Bell pair, which is not a linear mapping.

A short mathematical derivation: let $|\psi\rangle = a|0\rangle+b|0\rangle$ be a qubit state; $|\psi\rangle|0\rangle = a|00\rangle+b|10\rangle$ is a tensor product of $|\psi\rangle$ and $|0\rangle$.
[a,0,b,0] => [a^2, ab, ab, b^2] is neither a linear mapping not unitary.

Quantum communication, because of the no-cloning principle and decoherence, is practically instantaneous. This makes error correction a lot harder.

Quantum Money

Quantum currency is a multiple qubit state which can be used instead of classical currency. As they cannot be cloned it's impossible to have counterfeits, unlike classical currency. People can carry them around, trade them, etc. They can be verified by the bank, but at the cost of a qubit. However, at this point, it's strictly theoretical as its infeasible to store qubits, let alone carry them around.

Quantum Key Distribution (QKD)

When digital information is transmitted using quantum protocols, the uncertainty principle gives rise to novel cryptographic potential thats classically impossible as it's impossible to eavesdrop without a high probability of disturbing the state. This can be used to distribute random key information between users and ensuring confidentiality.

As described, quantum mechanisms hold a unique advantage over classical ones which leads to some interesting applications in quantum communication and networks. These can effectively address the spam and security problems listed earlier.

Quantum Communication Protocols to address classical [network] problems:

Spam (SPIT)	Authorised quantum-based access	Teleportation (Entanglement)	QKD
DoS attack	Authorised quantum-based access	Teleportation (Entanglement)	No-cloning principle
TCP SYN flooding	QKD	Entanglement timeout	
Remote Quantum Voting	QKD	Entanglement	Superdense coding
Data breach	Authorised quantum-based access	QKD	
Privacy	OKD		

Design

Design Flow

For the design, I follow two main papers: [1] and [2]. [1] gives a bottom-up, fairly realistic, algorithmic, low level stack and protocol design and implementation framework. It takes into account the quantum hardware strengths and limitations. [2], however, is theoretical, but gives an elaborate and well thought out design to build upon. This is top-down design approach. These two, therefore, give diverse perspectives and design considerations to build upon.

I've also incorporated a fairly polymathic design approach. I've retained the central concepts in classical networking - network stack, packet (datagram) design, layer protocols - and included several diverse design concepts from Operating Systems, Distributed Computing (HDFS), Programming Languages, Compiler design, making it a very effective and rich design.

I've focused on designing a quantum network framework which supports multipartite entanglement between network nodes. Different entanglement types - bell pairs, multipartite (GHZ, W, etc.) - provide different benefits, but at added costs of resource consumption. I use this differentiation to design a network architecture which uses specific entanglement types for specific applications. I've also designed a framework which allows dynamic conversion between entanglement types depending on currently available quantum resources and the protocols.

My design is broadly divided into two aspects: *Quantum Computing packages* and *Network Stack design* and *Packet Design*. Modularising the design in this sense leads to a more resuable, robust and an effective design. Every layer uses different Quantum Computing packages based on its responsibilities. Although, most of the Quantum operations will happen in the bottom 2 layers, as that made the most design sense. As outlined in [1], the functioning of the network will depend on these quantum operations, which depend on the underlying quantum computing hardware and its capabilities. Although the design will depend on quantum computer, it is fairly versatile which enables it to be configured it to different ecosystems as well.

In terms of network layers, my focus is on the lower layers in the network stack and in terms of routing, it's intra-domain networking. While I try to make use of both the approaches from [1] and [2], it seemed prudent to design lower layers first. Regarding the intra-domain networking: There are many organisations - academic and commercial - which are working on different aspects of quantum computing simultaneously. It would be logical to deduce that each will develop their own version of Quantum Autonomous Systems and network. Therefore, just as the first designers of the internet, the existing quantum network design must support such diversity in quantum computing.

However, within this boundary, things will be, more or less, homogeneous. As a result, it only seems prudent enough to design layers and protocols for intra-domain networking first and then address the inter-domain aspect later.

Design considerations

Design Challenges

1. Quantum Computing technology

The current state of quantum technology makes it difficult to develop a network framework at par with the classical network. The fidelity of the generated entanglement channel and the information qubit(s) is inversely proportional to the distance travelled, and the duration of storage. In addition, errors also creep in when qubits are generated. This effectively limits the applicability of quantum computing for communication. Given the laboratory conditions under which a lot of the experiments and research in this domain has been conducted, it further limits the real world applicability.

2. Cost

Currently, it isn't cost effective to develop a quantum computer, let alone have an elaborate network. The penetration to the degree of the current mobile technology and internet is a long way away. However, this also adds an advantage as restricted access does make it more secure.

3. Availability

This is one of the biggest challenges of quantum computing technologies. With the stability and fidelity of the currently generated qubits, the subsequent purification and error correction mechanisms reduces the availability.

4. Robustness

The no-cloning principle in addition to current qubit storage technologies make it extremely hard to ensure robustness. In case of failure, it is extremely hard and expensive for a node to reinitiate. The instantaneous nature of quantum communication makes it impossible to have technologies such as CDNs, caching, and all related technologies which aim to free up bandwidth by storing copies of traffic across multiple servers.

5. Implementation

Large scale implementation is a challenge. Although, Europe is really progressing in this regard with a pan-European Quantum Network initiative. Will there be exclusive pockets of Quantum Network ecosystems (like classical autonomous systems)?

Classical - Quantum Integration (conversion): As it may be infeasible to have a quantum computer everywhere, the reality is that there may be frequent Classical - Quantum conversions. This has to be made feasible, which again depends on the quantum computing technology. There are many concerns that need to be addressed. How will it be integrated with the classical network and to what degree? it is reasonable to have Quantum computers at critical/ strategic institutions and locations and allow others to connect and use these.

6. Performance

As mentioned earlier, quantum network needs to be instantaneous. Given that qubits and entangled state generated need to purified, it adds a layer of complication and delay, tightly binding throughput. In addition to throughput, error correction capacity and rate, resource management, information loss, all carry higher importance than they would do in classical networks.

7. Mobility

One of the strengths of classical internet is its reachability and support fo mobile devices, dynamic ip addresses. This is a big challenge for quantum computing. However, it would be worthwhile to asses if quantum computing would be useful in this scenario.

Design goals

When the classical internet was designed, as illustrated in [5], they had several design considerations which influenced the design of the architecture. Some aspects where classical network struggles and quantum solutions provides an elegant solution are:

1. Confidentiality

This is one quality (among others) which results in a secure networking framework. The very nature of qubit and teleportation mechanism makes this possible.

When an entanglement channel is established between two nodes, then any qubit teleported between them can only be decoded by them, as the teleported qubit is entangled with the channel. Therefore, any teleported qubit, if intercepted, cannot be decod-

ed without the other half of the entangled channel. This results in a highly confidential communication.

2. Integrity

The very nature of a qubit makes quantum communication highly integral. Information stored in a qubit isn't known until it's measured. Measurement causes the qubit to collapse to one of the base states. Only, we don't know which one. The only way to decode the original qubit is through the teleportation mechanism. So, if anyone tries to sniff the qubit it collapses and the information is lost, thus resulting in a highly integral communication.

3. Accountability

As [1] illustrates, the heralded entanglement mechanism provides a very reliable way to establish channels between nodes with accountability. And given the transparency in the teleportation mechanism, it results in a highly accountable network.

4. Authenticity

As mentioned in the cases of spamming and remote voting, one of the problems is verifying the authenticity of the requests. Given the no-tampering and no-cloning property of qubits, they can provide elegant, albeit complex, solutions.

5. Variety in Quantum Computing Communication protocols

The classical internet was designed to support multiple types of communications, as the designers didn't what would run on the Internet. However, in Quantum Networks, there are various protocols - Teleportation, Superdense coding, QKD, CHSH, etc. - all of which depend on the underlying fundamentals which enable information exchange - entanglement and qubit transfer.

Entanglement plays an important role in enabling information transfer. Therefore, Irrespective of the type of entanglement and the quantum technology used to generate it, functionally it will remain the same. My design is with *functionality as the variable.* It's prudent to design a well engineered entanglement-based network despite the differences in the quantum technologies whilst also supporting the different protocols.

Given the advantages the multipartite entanglement hold [6], it makes logical sense to design a network architecture which supports this and various. However, given how resource expensive it is to generate them, it might not be feasible to establish a multipartite entanglement channel between nodes at all times. In such cases, the network must dynamically adopt and convert between entanglement types or other protocols.

With this in mind, I have arrived at a design which enables dynamic and instant conversion between entanglement types, depending on the available quantum resources, protocols, applications etc.

6. Intra-domain networking

Fortunately or unfortunately, We find ourselves in a situation similar to when the classical internet was touted all those decades ago. As [5] illustrates, one of the design goals was to enable integration and communication between autonomous systems which were configured differently. Today, there are multiple entities - corporate, academic and government - which are working on different types of quantum computing technologies. There is the superconductor-based qubit generation, NV based, topological qubits etc. It is only logical to deduce that each will develop its own quantum computing ecosystem and we need to design a network framework which would enable communication and integration between these ecosystems.

Focusing on the intra domain networking provides us with an added benefit: although I'll have to stick to a particular quantum ecosystem, it will in contrast enable me to arrive at a more open ended design, as the communication is based on entanglement and teleportation.

Quantum Computing components

Entanglement channel

Entanglement is arguably the most important component. It is first attempted between the participating nodes, over which information (qubits) is teleported. This would be responsible for establishing entanglement links between nodes. The type of entanglement would depend on the application, protocol, available resources etc. It would essentially be an API (a class or a module), which would accept arguments such as entanglement type, participating nodes, priority etc. This would also be a timeout.

Purification

Given the error in the system, the fidelity of the channel must be over a threshold to ensure the quality of information. This is where purification is performed on the generated entangled bits (ebits). Once an entangled state with a minimum fidelity level is generated a heralding signal is generated which is then transmitted to the participating nodes, following which the respective qubits are also sent. Now, the channel is ready to teleport information qubits.

This would composed subcomponents: entanglement generation, increasing purification, generate heralding signal, swap entanglement with further nodes and teleportation of qubits.

Quantum computing packages

This is where the rubber hits the road. This is the prime module which performs all of the quantum computing operations, as this is the only module which is connected to the underlying quantum computer and has access to its resources. All of the other modules must call upon the APIs provided by this module to access quantum resources and computing operations.

This effectively abstracts away the underlying variations in quantum computing technologies. The APIs would remain the same but their implementations would differ depending on the quantum computer/ technology. Thus, once designed, developed and refined, the whole networking architecture would remain the same and theoretically could be replicated across different quantum computing ecosystems.

As an example, I could implement this module with SImulaQron simulator and design the whole architecture. Then, if I decide to change the underlying implementation using a different simulator, it wont affect the network design in any way. Only, the performance and metrics would change.

This component performs some of the basic quantum operations such as:

1. **Generating qubit**

This submodule provides the API which is to be used to request a qubit from the underlying quantum computer. This API call triggers the computer and once a qubit is generated, it returns with a pointer to it.

2. **Generating Entanglement**

Once you have the qubits, generating ebits is a sequence of unitary operations which entangles the participating qubits. So, this module would invoke operations in the Unitary Operations submodule.

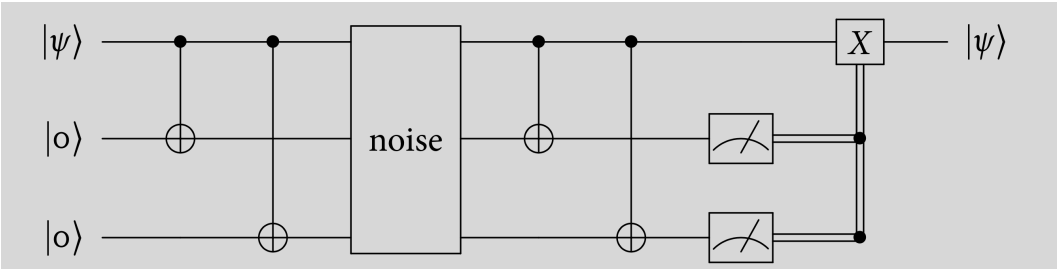
3. **Entanglement purification**

[3] provides a purification scheduling algorithm to increase the fidelity of the generated entangled state. While this is a fairly straight forward implementation, it is resource expensive. I have yet to explore different purification mechanisms.

4. **Quantum Gate operations**: Unitary transformations (Hadamard, X, Z, etc)

5. **Qubit error correction (QBEC)**

Two additional ancilla qubits are sent with eh information qubit. The noise affects all of these. Upon receiving them, the information qubit can be recovered as in the circuit.



QBEC circuit design with 1 info and 2 ancilla qubits

Quantum nodes

This is an important abstraction. This makes use of the various modules and provides an abstraction whose functions are a heterogenous combination of:

1. **Quantum computing** - This includes performing necessary quantum computing operations, quantum communications

2. **Networking** - Some degree of decision making like dynamic routing, packet (datagram) creation, protocol implementations and enforcement, network quality, establishing entanglement links based on protocols etc.

3. **Miscellaneous** - Maintaining the *node* and *network* state which consists of the available quantum resources in terms of entanglement types, entanglement generation capacity, any issues or errors during operation.

These are of different types: end nodes, intermediate nodes: automated (dumb) nodes and intelligent-master nodes, which are described below.

1. **End hosts** have very limited quantum networking capabilities: sending and receiving qubits, generating entanglement channel and . These do not have have any routing decision making abilities. Their quantum computing capacity is also limited and related their networking abilities.

2. **Automated intermediate nodes**, as illustrated in [1], have minimal or no decision making ability, although have a higher quantum computing capability. Their primary responsibility is to teleport the received qubit as per the received routing decision. One of the changes to the quantum packet here is that it strips the network state header of its sender and attaches its own.

3. **Intelligent-master nodes** have more decision making abilities and higher quantum computing capabilities than the automated nodes. The latter often report to these nodes in case of anything: sending frequent heartbeats, reporting in case of any issue etc. They receive regular heartbeats from all the other nodes about their network state, which are all collected and further transmitted to the master cluster.

This also has certain dynamic routing decision making ability, which is determined by the available quantum resources, entanglement conversion, etc. based on the updated network state. If all is good it teleports the received packet as is. If not, it makes an ad-hoc route change.

Quantum Network state

This state is divided into **private** and **public**. The private node state is the primary state of every node, which consists of details about the quantum computer, internal operating capacity, and any issues and bugs of the (sub)modules of the quantum computing package. The private isn't visible to any entity outside of this node.

The public state is abstracted atop the private state. This state is transmitted to its neighbours through every quantum packet transmission included in the datagram, and to a nearby master node frequently in the form of *heartbeats*. Only the node's neighbours and some intelligent nodes are privy to even the public version of the node state.

The *Node State* consists of, but not limited to:

1. Qubit creation capacity
    1. Number of qubits
    2. Minimum fidelity
2. Entanglement generation capacity
    1. Number of qubits in the entangled state
    2. Entanglement type
    3. Minimum fidelity
3. Entanglement Purification
    1. Scheduling
    2. Estimated time
    3. Minimum fidelity
    4. Rate of purification
4. Error correction
    1. Capacity
    2. Rate
5. Transmission rate

The network state also consists of the Entanglement Channel States between the nodes. The channel metrics are illustrated in detail later on. The Entanglement channel state consists of, but not limited to:

1. Entanglement Type - Bell pair, GHZ, W, etc.
2. Participating nodes
3. Fidelity at creation
4. Decoherence rate
5. Propagation rate/ channel capacity
6. Creation/ Expiry timestamp

Quantum Operating System (QOS)

Every node, as mentioned before, is a collection of heterogenous operations, from quantum computing, to quantum networking, to classical networking, to node state maintenance, and other classical operations. Furthermore, these different aspects are inter-dependent: the node state depends on the available quantum computing resources, routing and teleportation also depends on the same. To handle all of this cohesively and seamlessly there needs to be a comprehensive software for every node.

These nodes must also be reachable remotely to pull updates and push upgrades. There needs to be some generic, but secure APIs which accomplishes this. The design of the modern operating systems are good enough to be carried over, in a fair capacity. But, they must be modified to support the nature of the quantum computing technologies, which are starkly different from the classical computing technologies.

Distributed Quantum Resource manager

As mentioned before, every node maintains a state, a public version of which it sends encoded in the quantum packet. The intelligent-master nodes collect the states from surrounding nodes via frequent heartbeats and updates its local network state. This updated local state is then transmitted to the master cluster which updates the network state of the entire quantum computing ecosystem. This is similar to distributed computing framework as implemented in HDFS. The master cluster and/or intelligent-

master nodes will ping individual nodes if they haven't received heartbeats from them in a while. The network state is dynamic and updated.

Quantum Network stack and networking protocols

[1] and [2] have provided a good design base and approach to build upon. I've tried to incorporate both into my design, as appropriate. [2] has provided a good abstractive stack and descriptions of the layers.

With the design considerations I needed a relatively open network stack from the bottom up, which could be implemented across different quantum ecosystems, hopefully with minimal change. In this regard, each layer on the network stack would essentially be a set of fairly generic APIs and protocols.

My focus in this research will be limited to intra-domain networking which corresponds to the lower layers: physical layer, connectivity layer, link layer and network layer. Most of the operations related to quantum computing would be part of the physical layer. This would interact with the underlying quantum computer - different for every ecosystem - in generating qubits, entangled bits and performing quantum operations.

1. Physical

Goal: Establish physical connect between quantum networking devices, perform necessary quantum computations
Functions: Quantum Computing operations, Generating qubits, generate entangled state, qubit error correction, entanglement purification, entanglement swapping, heralded signal generation

2. Connectivity

Goals: Ensure long-distance P2P connectivity, Node State, Repeaters
Functions: Entanglement generation scheme, Node state maintenance
Quantum repeaters, Long-distance P2P entanglement, Entanglement distillation

3. Link

Goals: Generate network state for a network, Intra-network routing, Switching
Functions: Generate arbitrary network states, Network state maintenance - Node states, adjacency matrix, Switches, Intra-network routing, communicate to connectivity layer, P2P entanglement, communication protocol conversion

4. Network

Goal: Inter-network communication
Functions: Routing between Network states, Boundaries - quantum-classical network, bipartite-multipartite, different Quantum Computing ecosystems, Router, Intra-domain decision making, Share network states - internally and externally, communication protocol change

Quantum packet (datagram)

Much of the datagram will remain the same. However, with another added layer (connectivity), some changes will be incorporated. As I mentioned earlier, the node state is also included in the datagram when the quantum packet is teleported to the neighbouring node.

Implementation

Design

In the design framework elaborated previously, with regards to the network stack, I've implemented the *Physical* and *Connectivity* layer to an extent and yet to implement the Link and Network layers and the Protocols.

With regards to the components, I've implemented to an extent, Entanglement channel and Quantum computing packages. I've yet to implement the quantum node abstraction, node and network state, Quantum OS and quantum distributed resource manager. The latter two are out of scope of this research, as they need a much deeper research into Operating Systems.

All of the design and simulation implementation has been programmed in Python.

Simulation

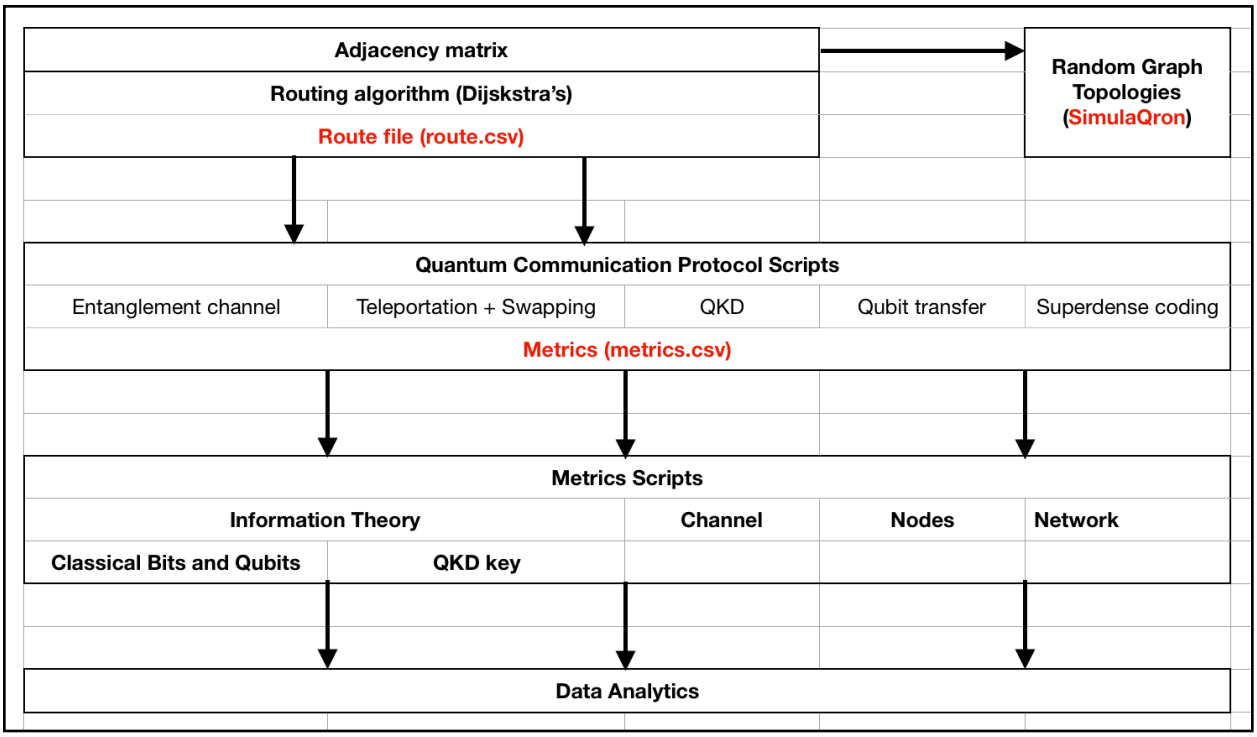
SImulaQron supports 2 ways of simulation: One is to use the native interface directly and there other is to use the classical-quantum combiner (CQC) interface built on top of this. I've used both, but largely CQC. The CQC library has pythonLib, which provides APIs that are simple and straightforward, leading to a cleaner code.

I've implemented a slew of protocols - qubit transfer, entanglement channel, teleportation, entanglement swapping, superdense coding, QKD, QBEC, across different topologies - dual (1:1), line, random graph and voting (N:1) - with and without system noise leading to varying error rates in bits, qubits and entangled state.

Adjacency matrix, Random Graph and Dijskstra's routing algorithm
The implementation of the protocols is fairly robust and is independent of the topology. The protocol scripts require a csv file containing the route (*route.csv*) - a list of node ids - and the simulation runs with that route. This route can be linear or a random graph. For a random graph topology, it needs an adjacency matrix as input. A route is determined by Dijskstra's algorithm, which is saved into a csv file. This is then fed into the protocol scripts.

The protocols generate data - qubits/ bits (bit flip error rate, entropy), channel (noise rate, purity level), nodes (qbec rate, qubit capacity) - all of which are saved into a raw metrics.csv file. The raw data files are fed into metric scripts which extract useful information. Data analytics can be applied on this processed data in visualisations.

My modular implementation of each of the abstraction provides high flexibility. I have used the same codebase for almost every topology and protocol.



Implementation Architecture

Topology	Description	Entanglement Channel	Teleportation	Entanglement Swapping	Superdense coding	Qubit Err correction (QBEC)	QKD
Dual	2 nodes 1:1	Establishing an entanglement channel between nodes for every qubit of information transfer	1:1 teleportation		Encoding classical info (bits) into Bell pair bit and transferring	Measuring qubit Error rate through N simulation runs	
Line	N nodes End to end Max degree: 2	Establishing an entanglement channel between nodes. Ad-hoc channel between nodes	Serial teleportation between nodes	Teleporting qubits swapping them between entangled channels through quantum operation	Serial transfer of bits end to end using Bell pair qubit. Measurement at each node	Measuring End to end & Node to node Bit flip error	
Random	Adjacency matrix Dijskstra's routing Dynamic, ad-hoc	Establishing an entanglement channel between nodes. Ad-hoc & dynamic route channel	Sequential teleportation between nodes by dynamic routing	Teleporting qubits through swapping between entanglement channels	Sequential transfer of bits using Bell pair qubit. Measurement at each node	End to end & Node to node Bit flip error based on node degree	Node to node QKD for critical applications Ad-hoc routing
Voting	1 (voter) :1 (Govt) N (voters):1 (Govt)	Establishing an exclusive entanglement channel between client (voter) and server (govt)	Teleporting qubits between Govt. and a voter		Encoding Votes (classical bits) into entangled qubit and transferring them	QBEC by voter for QKD qubit. QBEC BY govt for vote qubit	Generating Secret keys for every voter. Distributing them to voters

Implementation Summary

Metrics

Information Theory

1. **Entropy -** Calculate the loss/ gain in information when transferred, in terms of 1, 2 qubits
2. **Qubit error rate -** When a qubit is generated it isn't 100% pure, because of the noise in the system. When they're transferred between nodes, the noise in the channels also degrades them, causing bit flip errors.
3. **QKD key loss -** Bit flip error leads to loss in some of the qubits in a key. More is the % loss, higher is the threat probability.
4. **Threat estimation -** An estimation based on entropy, qubit error rate and QKD loss

Channel

1. **Degradation rate** - The fidelity of the generated entangled state isn't 100% and it also decoheres with time and distance
2. **Capacity** - Max number of qubits it can carry simultaneously.
3. **Propagation rate/ delay** - Currently, with the given quantum technology. This increases with QBEC.
4. **Throughput** - This is similar to the classical definition of throughput.

Node

1. **Qubit generation capacity -** Quantum Computers cannot generate unlimited qubits as it's as expensive process.
2. **Error rate** - The percentage of qubits generated with QBER > threshold
3. **Error correction rate** - Successful QBEC rate
4. **Transmission delay** - transmitting qubits between nodes depends on the underlying quantum technology, the established channel, QBEC, etc. The number of qubits in the entangled state that could be used for teleportation determines how many qubits can be teleported at once. The quantum technology can be atomic or concurrent with access to its quantum computing resources which also determines the transmission delay. As in the results, QBEC increases this delay a lot.

Network

1. **Network Fingerprint -** A unique identity vector, which is a combination of multiple metrics described earlier for every network
2. The network state is an average of all of these values. It also maintains account of the weakest and strongest node and link.
3. **Dynamic adjacency matrix** and topology changes
4. Supported applications and protocols

Results

Dual Topology

QBER | Dual Topology
Estimated QBER ranges between 0.05 up to 0.2 (from 60 simulation loops).

Qubit flip Err, Transmission delay | Dual | w/o QBEC					
Pair	# Err	% Err	Total	Avg Delay	Total Delay
0-1	21	0.127	165	0.208	34.381651878356934

Qubit flip Err, Transmission delay | Dual | QBEC					
Pair	# Err	% Err	Total	Avg Delay	Total Delay
0-1	28	0.184	152	0.242	36.85255432128906

The above results show that QBEC adds a delay in transmission. However, the adopted QBEC algorithm wasn't as efficient as expected.

Line Topology

Teleportation | Line Topology | 8 nodes | route [0,1,2,3,4,5,6,7]					
Node	**Sending Delay**	**# sent**	**Receiving Delay (ms)**	**# received**	**Total delay (ms)**
Node0	1.487687349319458	15	0	0	1.487687349319458
Node1	1.625908374786377	15	0.6520946025848389	15	2.278002977371216
Node2	1.724534034729004	15	0.7106857299804688	15	2.4352197647094727
Node3	1.4247162342071533	15	0.9964292049407959	15	2.421145439147949
Node4	1.4013340473175049	15	0.7990708351135254	15	2.2004048824310303
Node5	1.3217122554779053	15	0.5366230010986328	15	1.858335256576538
Node6	1.638268232345581	15	0.49966907501220703	15	2.137937307357788
Node7	0	0	0.9199206829071045	15	0.9199206829071045

Superdense coding | Line Topology | 8 nodes | route [0,1,2,3,4,5,6,7]					
Node	**Sending Delay**	**# sent**	**Receiving Delay (ms)**	**# received**	**Total delay (ms)**
Node0	0.14720916748046875	20	0	0	1.7469415664672852
Node1	0.1407780647277832	20	0.14088106155395508	20	7.745494365692139
Node2	0.1883375644683838	20	0.043022871017456055	20	5.533462285995483
Node3	0.16571807861328125	20	0.3704698085784912	20	13.59057331085205
Node4	0.16477036476135254	20	0.16021442413330078	20	7.759669065475464
Node5	0.14429497718811035	20	0.03491544723510742	20	4.035194635391235
Node6	0.1984238624572754	20	0.051971435546875	20	6.110128402709961
Node7	0	0	0.04402041435241699	20	0.45766711235046387

Random Graph

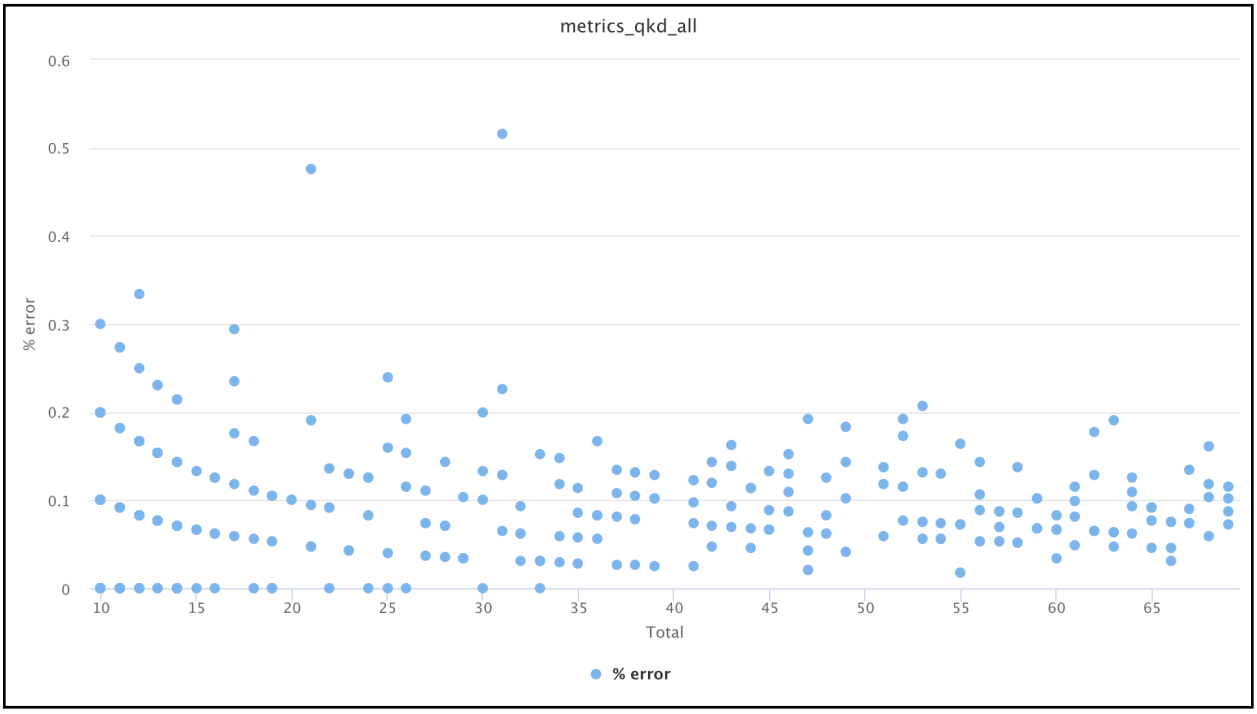
Adjacency matrix			
1	0	**2**	0
0	1	1	**2**
2	1	2	1
0	2	1	1

Superdense coding | Random Graph |			
Channel	**1st bit err**	**2nd bit err**	**Total transmissions**
0-2	0	0	10
2-3	0	0	10

Superdense coding | Random Graph | Node Metrics | route [0,2,3]					
Node	**Sending Delay**	**# sent**	**Receiving Delay (ms)**	**# received**	**Total delay (ms)**
Node0	0.041556596755981445	10	0	0	0.2245802879333496
Node2	0.03780961036682129	10	0.014616012573242188	10	0.5436899662017822
Node3	0	0	0.015557050704956055	10	0.09365320205688477

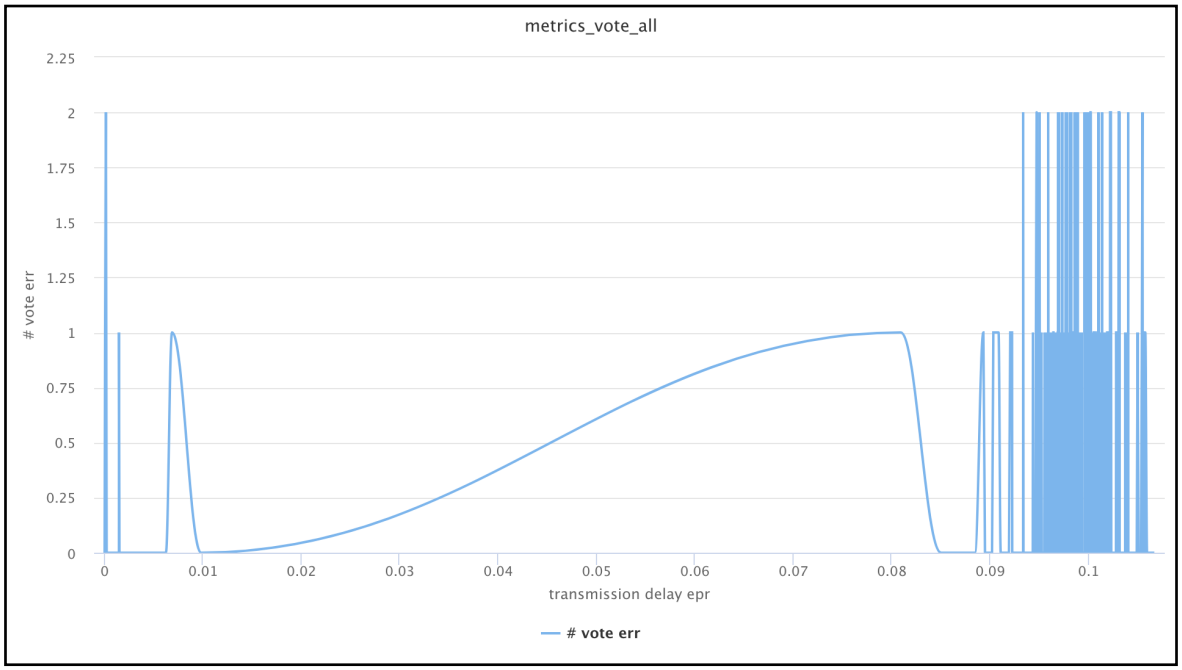
Teleportation | Random Graph | Node Metrics | route [0,2,3]					
Node	**Sending Delay**	**# sent**	**Receiving Delay (ms)**	**# received**	**Total delay (ms)**
Node0	2.226793050765991	20	0	0	2.226793050765991
Node2	2.3616766929626465	20	0.292694091796875	20	2.6543707847595215
Node3	0	0	0.8517935276031494	20	0.8517935276031494

Quantum Voting

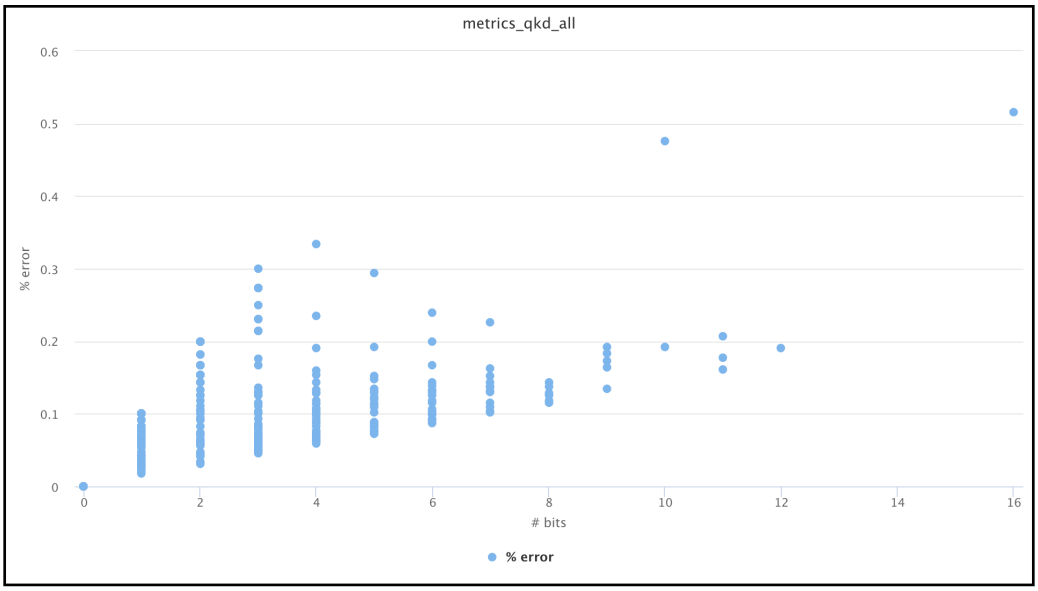


% error - Total key size for QKD generation

The qubit flip err rate ranges from 0-0.2. This reduces the effective size of the secret QKD key. It seems to stabilise as the QKD size increases.



 # vote bit flips - transmission delay. Bit flip errors increases with transmission delay.



% qkd err - # qkd err. % err seems to stabilise as # qkd bit err increases

Information metrics | Quantum Voting | 5 (voter) :1 (govt)
Entropy diff: 0.0011194518534580755
156 1 qubit flips
36 2 qubit flips
% loss of key is 0.09131323159655264
Threat from loss key and 1 qubit flips: 1.7456941334635063
Threat from loss key and 2 qubit flips: 0.4028524923377322

How delicate quantum voting can be. Even a single bit flip is unacceptable.

Conclusion

The Quantum Computing framework provides an elegant set of protocols which can be used to effectively address some of the classical network problems listed at the beginning.

However, based on the simulations, system and channel noise are some of the major obstacles. In critical applications such as voting even a single bit flip might result in a different candidate being elected. Qubits and entangled channels are very sensitive to noise, which makes them very secure but equally fragile.

Therefore, in such scenarios where the error margin is none, it would be good to implement advanced QBEC algorithms and use a multipartite entanglement protocols which offer better error correction [6], at a higher expense though.

This again ties back to my original research direction, to attempt to design a network framework based on multipartite entanglement.

Future work

1. **Elaborate simulations** - I'd like to explore various communication protocols with more elaborate simulations with complex topologies and concurrent networking.
   - Complex topologies
   - Refined metrics and data analytics
   - Better QBEC algorithm
2. **Multipartite entanglement** - GHZ, W
   - As mentioned earlier I would like to attempt a design which involves multipartite entanglement, as these offer better error correction.
3. **Quantum Network stack design**
   - Refine Design and Implement of Physical and Connectivity layers
   - Design Link and Network layers
   - Quantum packet design
   - Network state design and management
4. Simulation demonstration for the **remaining use cases**
   - SPIT, TCP SYN, Denial of Service, Data breach
5. Quantum - Classical integration
   - A Quantum - Classical convertor, protocol mapping

Glossary

1. **Bell pair** - An entangled state with 2 qubits (bipartite)
2. **CHSH** - John Clauser, Michael Horne, Abner Shimony, and Richard Holt's experiment proving that certain consequences of entanglement cannot be explained by hidden variables as stated by Einstein–Podolsky–Rosen paradox.
3. **GHZ** - Tripartite maximally entangled state
4. **W** - Tripartite non-maximally entangled state
5. **QBER** - Qubit Error rate
6. **QBEC** - Qubit Err Correction
7. **LOCC** - Local Unitary transformation and Classical Communication
8. **QKD** - Quantum Key Distribution
9. **Fidelity -** how closely a realised entangled state is to the ideal state.

References

1. https://arxiv.org/abs/1903.09778
2. https://arxiv.org/abs/1810.03556
3. https://arxiv.org/abs/0705.4128v2
4. https://doi.org/10.1038/299802a0
5. https://dl.acm.org/doi/10.1145/52325.52336
6. https://arxiv.org/abs/1909.00862
7. https://www.cvedetails.com/vulnerabilities-by-types.php
8. https://techcrunch.com/2020/04/13/coronavirus-vote-by-mail-wyden-klobuchar/
9. https://www.npr.org/2020/05/15/856189149/it-s-partly-on-me-gop-official-says-fraud-warnings-hamper-vote-by-mail-push
10. https://www.foxnews.com/politics/what-is-ballot-harvesting
11. https://www.foxnews.com/politics/lawyer-threaten-to-sue-nevada-unless-ballot-harvesting-is-permitted