Aliro™

Behind the Scenes: Building BrightNet, Aliro's Quantum Secure Communications Network

Aliro

www.alirotech.com

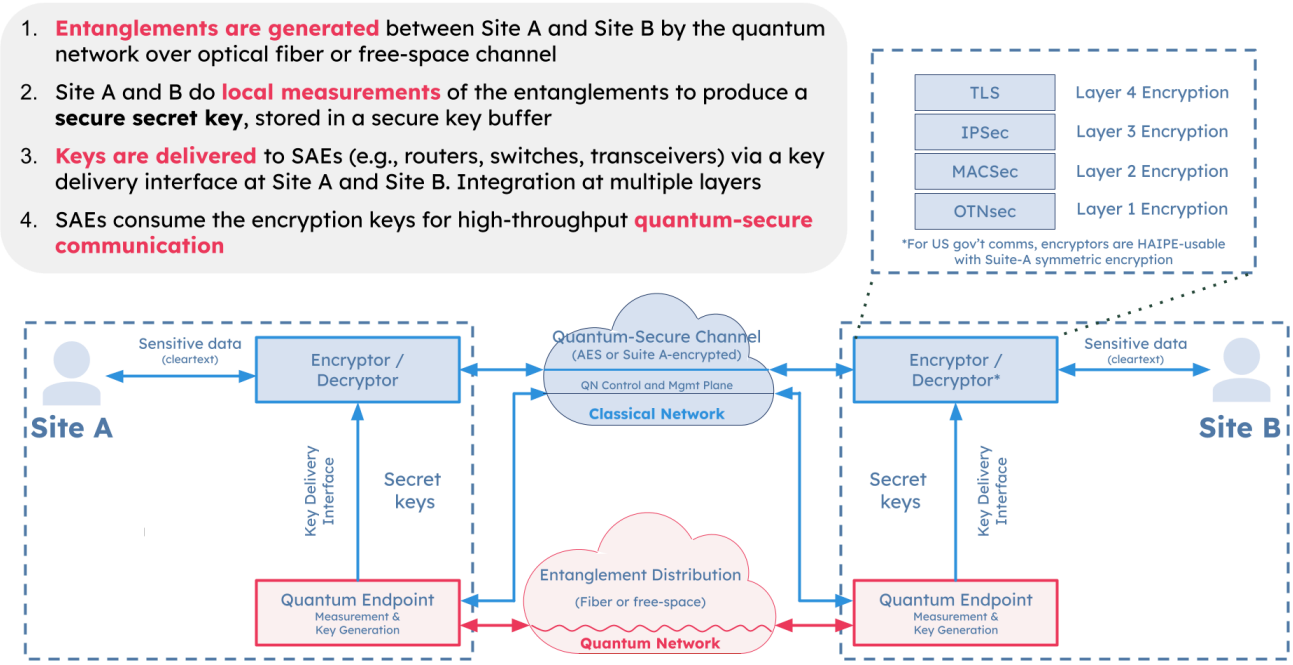
Behind the Scenes: Building BrightNet
Aliro's Quantum Secure Communications Network

Summary

This white paper gives an overview of the iterative simulation and validation process Aliro used to build BrightNet, highlighting the roles of quantum network simulation and emulation in designing, validating, and iterating on a working quantum-secure VPN network. BrightNet is a full-stack entanglement-based quantum network running production-grade secure communications services located at Aliro headquarters in Boston, Massachusetts.

Introduction

Every Internet connection between users, data centers, and clouds uses encryptors and decryptors. When two sites want to communicate, one site uses an encryptor to scramble the information they want to send, so that when this information traverses a classical network like the Internet, it is unreadable unless your decryptor has the key to unscramble it: to decrypt the information and recover the voice, video, or text message that was sent.



These devices need a constant source of fresh, high-quality keys in order to maintain the security of our communications as they are sent across the network.

Encryptors are prevalent and everywhere in today's existing network infrastructure and they require a constant flow of secure keys. Now the question becomes, how are these encryptors actually establishing these keys between Site A (sender) and Site B (receiver)? This is the primary security vulnerability that quantum key distribution solves. Essentially, the quantum network can be thought of as an underlay to the existing classical network infrastructure. It's an additional capability: the quantum network is generating quantum entanglement and distributing it to these two sites. Through the BBM92 protocol, or similar entanglement-based

protocol, entanglement is used to produce keys that are then fed up into the encryptor. All that changes is where the encryptors are getting their keys from. This process is Quantum Secure Communications (QSC).

Today, key exchange is accomplished out in the open over the Internet: the Internet is used to perform public-key cryptography using math-based protocols. There's a lot of mathematical structure in how this key establishment is done on classical networks. This structure can be exploited by adversaries to learn the keys that the encryptors have. However, the quantum network acts like a kind of out-of-band key establishment network that provides additional security benefits: not only is its security based on physics rather than mathematical principles, it's possible to detect an eavesdropper before any sensitive data is sent over the network. In fact, the keys being generated are derived locally, unlike today's math-based protocols and algorithms. Instead, quantum keys are generated through measuring entangled photons at the end nodes.

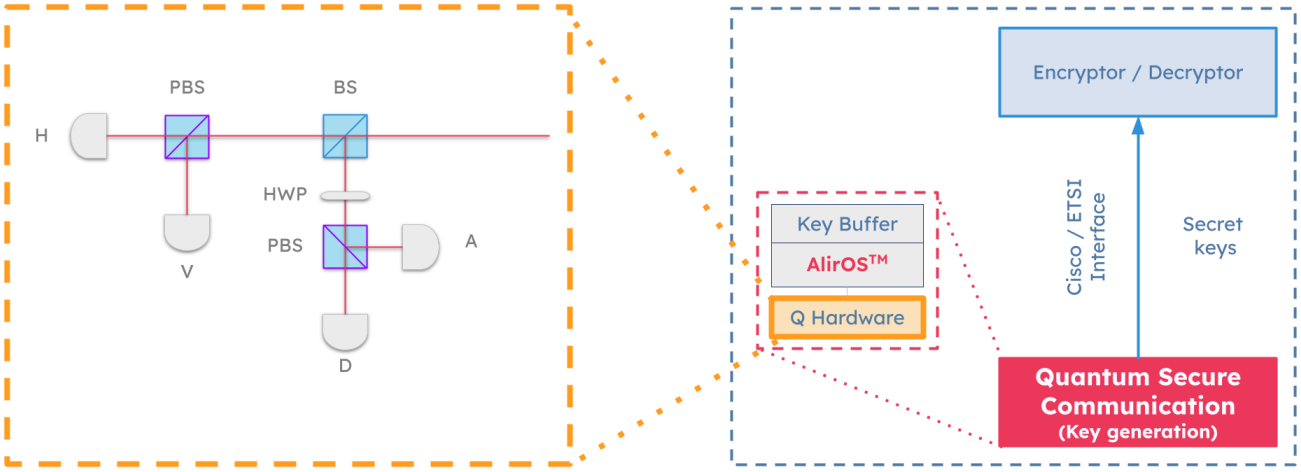
For more on how the process of generating keys using physics, please see the on-demand webinar Protocols for Quantum Networks. https://www.brighttalk.com/webcast/19861/612988

Just like traditional networks, quantum networks need software to abstract the complexities of managing quantum hardware and ensure reliable, scalable operation of the quantum network: an operating system that manages device behavior and can perform adaptive operations and a control plane to connect the quantum hardware and coordinate entanglement routing, monitor network health, apply traffic policies, and coordinate key generation.

The classical network and the quantum network need to communicate as well. This means the existing Network Management System and quantum network orchestration tools must interact to initiate key generation sessions, deliver keys, allocate bandwidth, and prioritize secure channels.

Once generated, there are many different integration points for actually using these quantum generated keys from the key buffers. The key buffers aggregate, store, and distribute quantum-generated keys to encryption systems that need them, whether that's IPsec, MACSec, TLS, and so on.

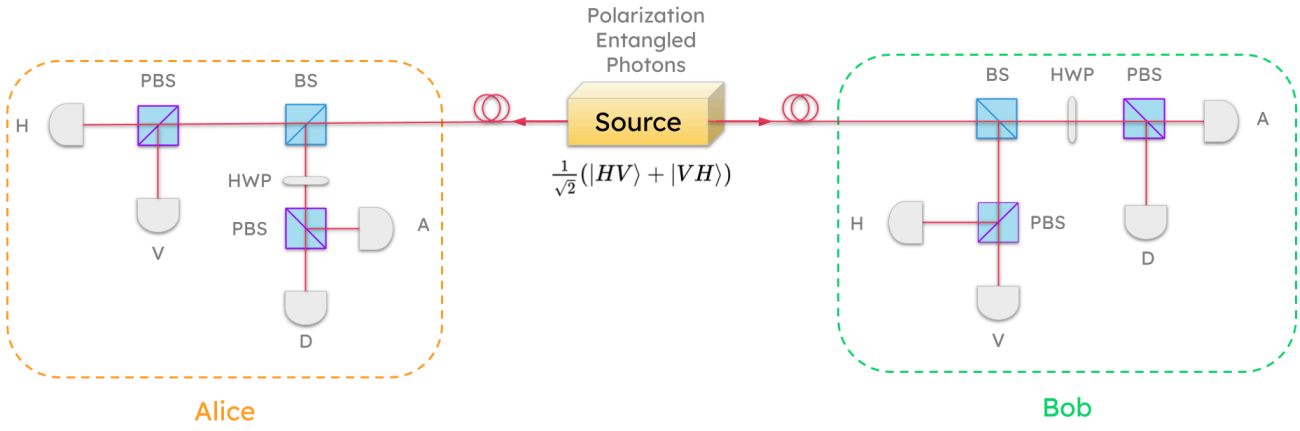
Let's zoom in on an end node in the quantum network from our previous diagram:



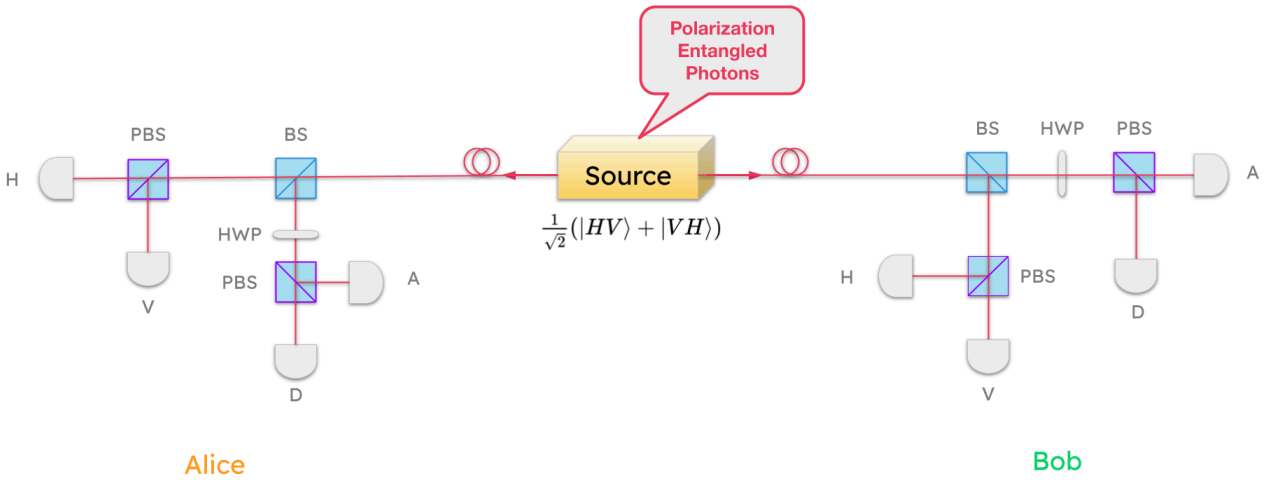
What does this look like in the real world? What are these "Quantum Secure Communication" boxes in the diagram? What do they consist of? To perform this key generation protocol, dedicated quantum hardware is required: photon detectors, photon sources, specialized optics, etc. These components make up the physical layer. This layer is the foundation of an entanglement-based quantum network and forms the core mechanism for Quantum Secure Communication (QSC). The security of this layer is grounded in the laws of quantum physics, not mathematics, making it tamper-evident by design.

A Day in the Life of an Entangled Photon Pair

How do entangled photons become quantum-secure keys with these components?

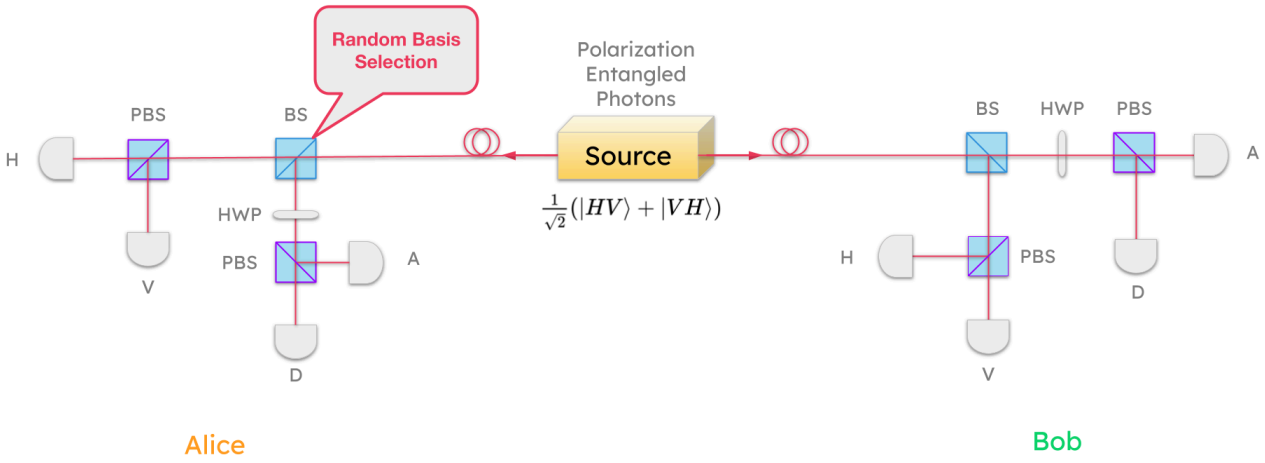


Pictured above is a hardware schematic for a polarization-encoded BBM92 network. This is BrightNet, a quantum network that was built at Aliro headquarters in Boston.



At the center is the entangled photon source. It's generating pairs of polarization-entangled photons at very high rates. One photon of the entangled pair is sent to Alice and the other photon of the pair is sent to Bob over some fiber. Alice and Bob have essentially the same set of hardware for measuring the quantum state.

Zooming in on Alice's node, one of the photons arrives at Alice and hits a beam splitter (labeled BS in the diagram). This beam splitter is a component that, as the name suggests, will split the beam of light that is coming into it from the source. For individual photons, this essentially means that approximately half of the time the photon will reflect off of the beam splitter and go towards the downward path. The other half of the time, the photon will pass through that beam splitter and go to the left. The beam splitter is a passive component that essentially injects some randomness in how the photon is measured.

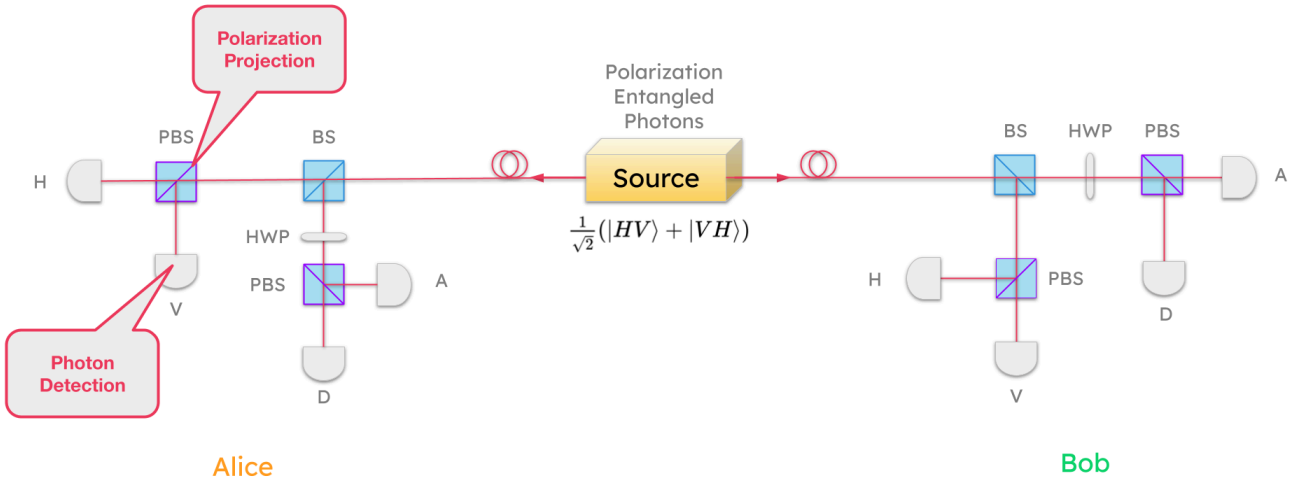


You can think of it as a prism that performs truly random basis selection for how we measure the incoming photon's polarization state:

- If the photon reflects, it goes to one basis measurement apparatus, the diagonal / anti-diagonal apparatus.

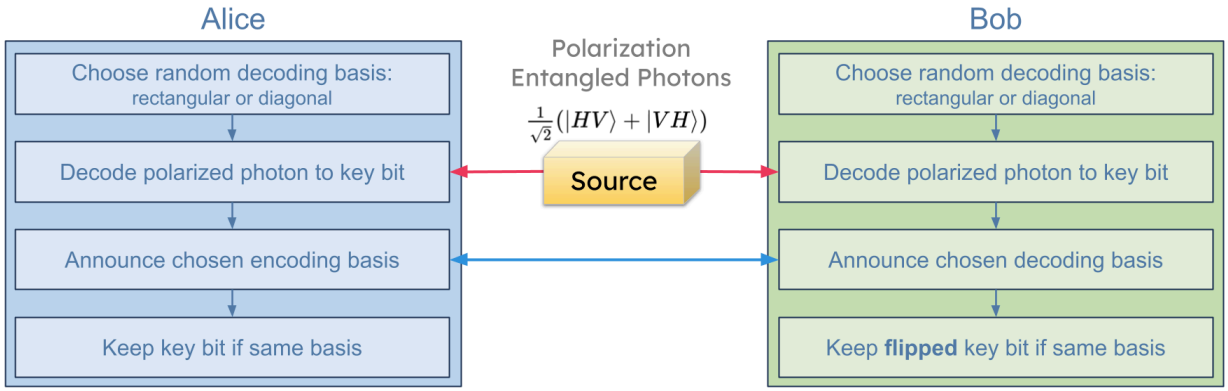
- If the photon passes through, it goes to a different basis measurement apparatus, the horizontal / vertical apparatus.

Once the basis is selected by the beam splitter (BS), the photon's quantum state needs to be projected onto a specific polarization axis.



In other words, once the beam splitter randomly routes the photon to one of two paths, we then measure it in that way, in that basis. This is called projecting the photon's state onto that axis. The polarizing beam splitter (labeled PBS in the diagram) and the detectors perform the projection automatically: the photon passes through the PBS, which has similar functionality as the beam splitter but it has a polarization filter that causes the photon to hit either the H detector or the V detector, at which point the photon's polarization has been measured and identified.

This same process is performed along the diagonal and anti-diagonal path.
That's the quantum portion of the key generation process, basis selection and measurement. Once the process of basis selection and measurement has been performed, the rest of the protocol is classical in nature, as the polarized photons are converted into classical bits.



All of these components in the quantum network come with certain profiles, certain physical parameters that determine their performance, their characteristics, the types of qubits they are compatible with, and what they can do with those qubits.

Physical Parameters

Measured Parameter	Value
Source visibility	94%
Source wavelength	810 nm
Source bandwidth (FWHM)	3 nm
Source rate (cps)	1.49E6 cps
Detection resolution	690 ps
Detector dead time	45 ns
Detector maximum efficiency	0.60 nm
Detector dark counts (Alice)	500 cps
Detector dark counts (Bob)	1800 cps
Alice link loss	12.0 dB
Bob link loss	12.0 dB

Given all these complex parameters, how do we build a network of compatible components that will perform at the fidelity and rate we require for a use case like Quantum Secure Communication? Quantum network simulation is fundamental to the implementation of quantum networks that operate as intended, by design.

Quantum Network Simulation for Design and Implementation

Having a simulator framework that supports the design and validation of the network, and can be used to predict expected performance metrics, is crucial to the success of any quantum network. Aliro used its own simulator to design BrightNet. Iteration of the design within Aliro Simulator using Aliro's tested simulator framework supported a smooth implementation of this quantum network, and once built, it performed exactly as designed and expected with the performance metrics required for Quantum Secure Communication.

Aliro's Quantum Network Simulator Framework

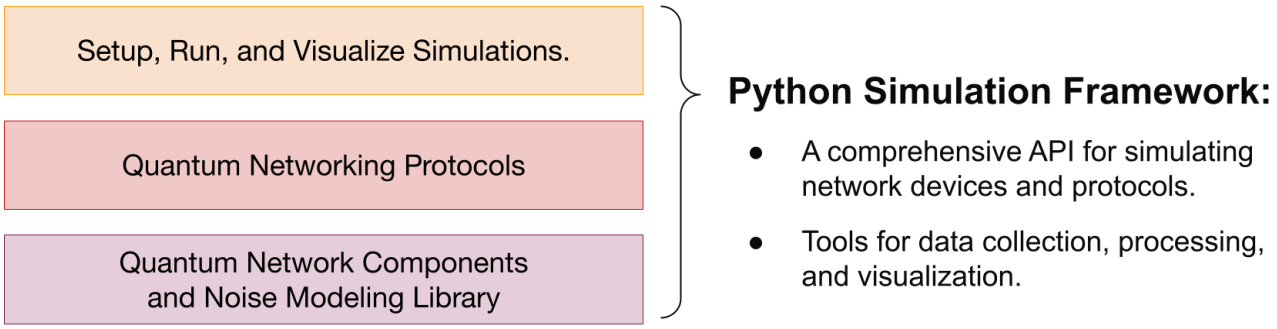
Aliro's framework used two distinct pieces: a Python simulation framework and a scalable, swappable backend.

A Python Simulation Framework.

A comprehensive API ties together 3 quantum networking simulation inputs:

- Quantum network hardware components.

- Noise modeling.
- Quantum network protocols.



Inside the Python portion of Aliro's framework is the Quantum Network Components & Noise Modeling Library. This is where every device's profile is collected, with details such as center wavelengths, spectral width, brightness, detector efficiency, timing jitter, dead time, max count rate, fiber/link loss, and component insertion loss, as well as noise models that are inherent in the specific fibers your network will use. In short, this library turns hardware datasheets and fiber characterizations into a digital model of the physical layer components.

Also in this part of the framework are the Quantum Networking Protocols. On top of the physics, the simulator runs the protocol logic Alice and Bob need: basis selection, sifting, error correction (with chosen efficiency), and privacy amplification. This lets you test not only if the optics interoperate, but whether the whole BBM92 workflow will meet targets (coincidence rate, QBER, secure key rate).

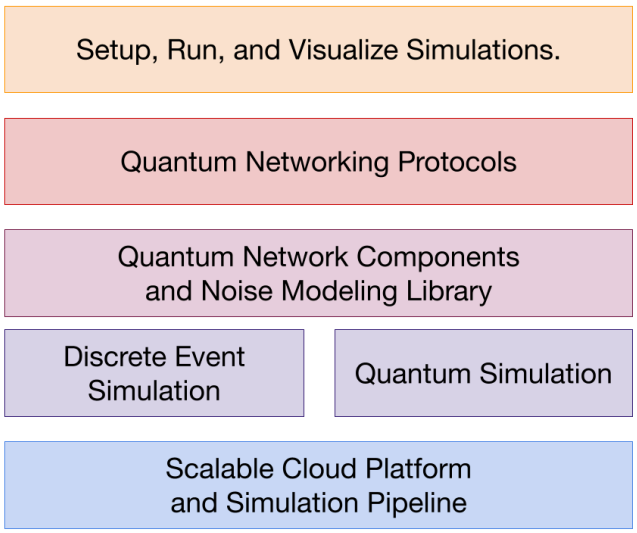
These capabilities support the operator's experience of setting up, running, and visualizing designs:

- Describing devices and fibers.
- Composing links and networks with precise parameters.
- Collecting and aggregating results.
- Visualizing coincidences, QBER, and secret key rate vs. distance or time, and many other performance metrics.
- Exporting configs to digital twin emulators and orchestration layers.

This part of the framework predicts how the components of the network will work together based on real-world parameters (wavelengths, filters, detectors, fiber) before any hardware purchases are made.

A Scalable, Swappable Backend.
Under the hood, you can pick the quantum state representations you need (state-vector, density matrix, stabilizer) and even swap simulators (Cirq, QuTiP, QSim, etc). This keeps the

simulation framework flexible for different needs and use cases, and supports a scalable cloud platform and simulation pipeline to provide parallel processing or GPU acceleration.



Python Simulation Framework:

- A comprehensive API for simulating network devices and protocols.
- Tools for data collection, processing, and visualization.

Scalable & Swappable Backend:

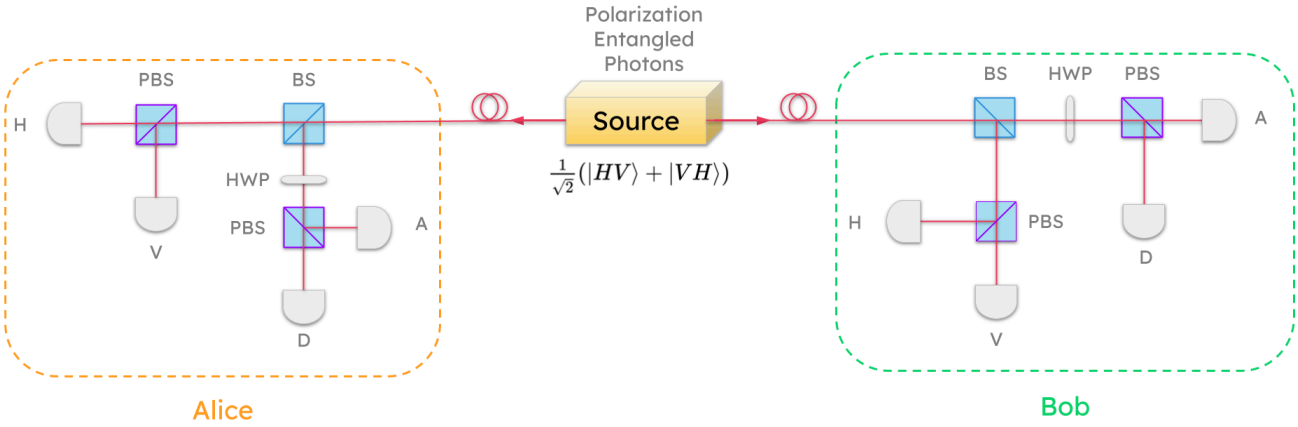
- Integrate with any quantum simulator (Cirq, QuTIP, QSim, ...)
- Supports multiple state representation (State Vector, Density Matrix, Stabilizer,...)
- Scale on the cloud using parallel processing or GPU acceleration.

Part of this backend is Discrete-Event Simulation. Discrete-Event Simulation models a system as a sequence of instantaneous events (arrivals, starts, finishes, failures). The virtual timeline is simply moving from one event to the next, allowing for ultra-precise time resolution. Behavior is driven by queues, resources, and stochastic distributions (e.g., arrival times).

The other part of this backend is the Quantum Simulation. This part of the simulation framework evolves the quantum states (e.g., $|HV\rangle + |VH\rangle$) through realistic fiber and optics, accounting for polarization rotation, dispersion, filtering, and beam splitters at picosecond or nanosecond resolution. The goal here is to realistically predict how a quantum state degrades on a real link and what that implies for visibility, QBER, and the key rate.

Putting it all together

So we have our proposed design.

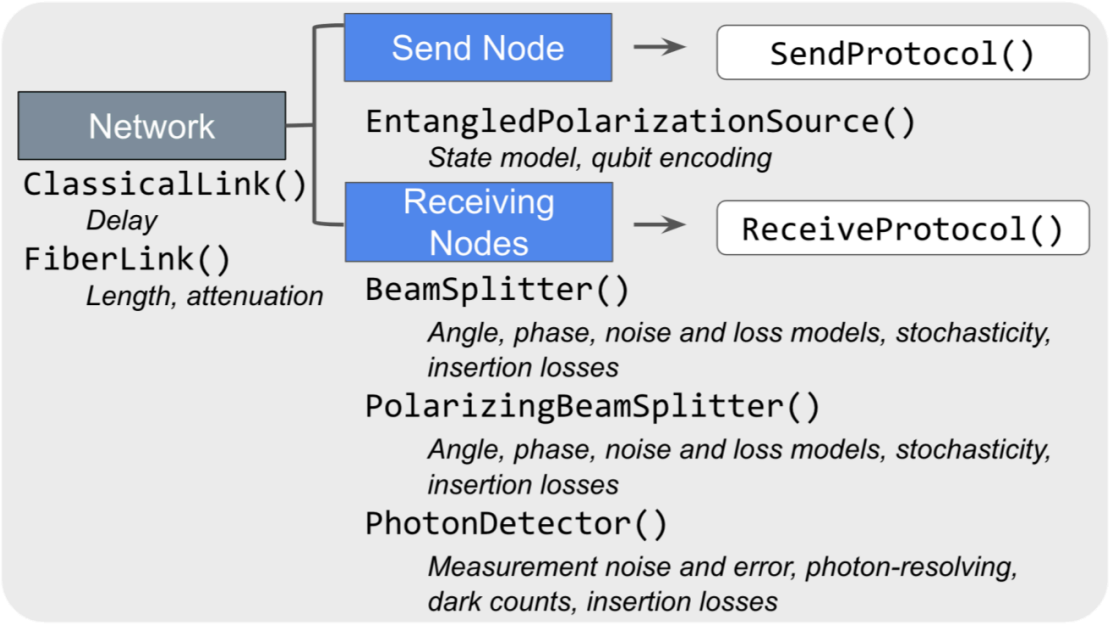


We have parameters and characterizations of components.

Physical Parameters

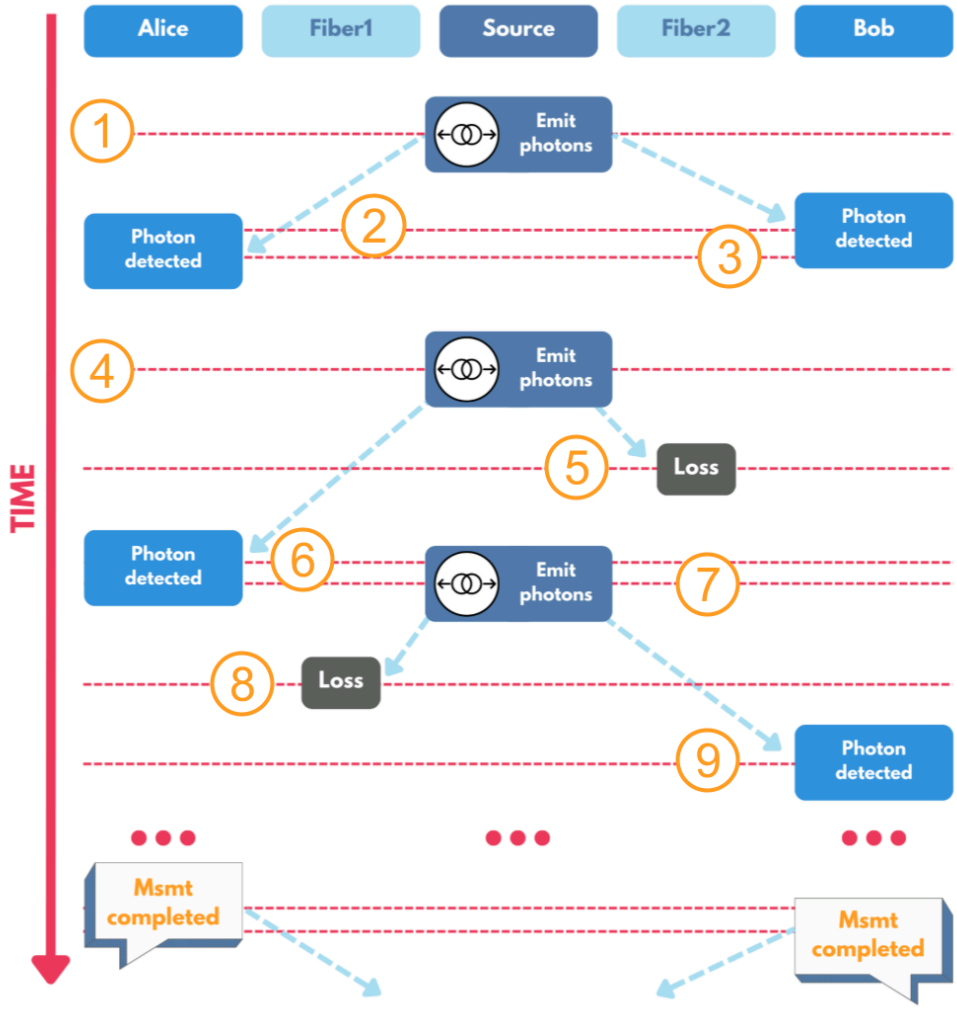
Measured Parameter	Value
Source visibility	94%
Source wavelength	810 nm
Source bandwidth (FWHM)	3 nm
Source rate (cps)	1.49E6 cps
Detection resolution	690 ps
Detector dead time	45 ns
Detector maximum efficiency	0.60 nm
Detector dark counts (Alice)	500 cps
Detector dark counts (Bob)	1800 cps
Alice link loss	12.0 dB
Bob link loss	12.0 dB

These pieces are then collected into a simulation of the quantum network, one that is highly accurate in simulating both the ideal scenario as well as the likely real-world performance of the network. Pictured below is a diagram of BrightNet, a Quantum Secure Communications link between 2 nodes, and some parameters related to the components in the network that have been incorporated into modeling performance.



Designed, verified, & optimized with Aliro Simulator

Pictured below is a simple visualization of the events occurring in a quantum network simulation. The red dotted lines represent a point in time that an event occurred.



1 - The Entangled Photon Source emits a pair of entangled photons.
2 and 3 - The photons arrive, and each photon is detected at Alice and Bob.

This process is repeated many more times to build a secure key.

4 - The Entangled Photon Source emits a pair of entangled photons.
5 - This time, the photon intended for Bob is lost.
6 - Alice's photon arrives and is detected.
7 - The Entangled Photon Source emits a pair of entangled photons.
8 - This time, the photon intended for Alice is lost.
9 - Bob’s photon arrives and is detected.

This process continues until a suitably sized string of secret bits is created at each end node.

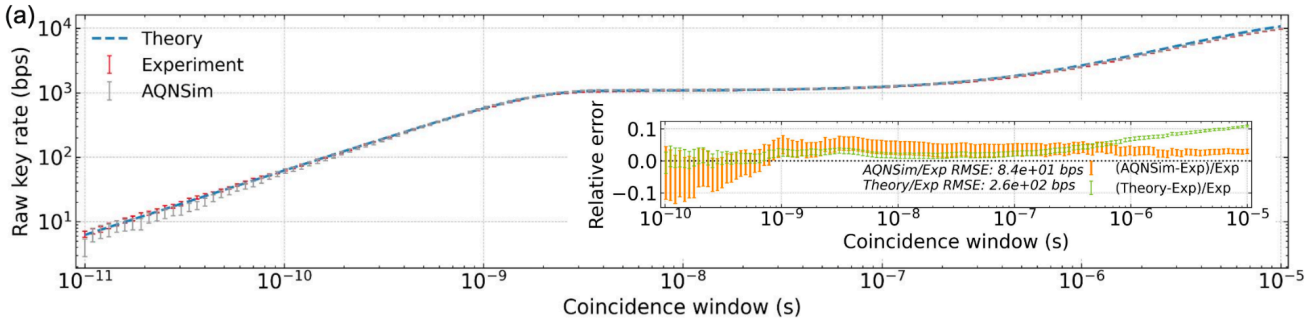
This is how the real network behaves in real operation, rather than the ideal scenario. This Monte-Carlo discrete-event framework is really helpful for modeling reality, and it's a virtual timeline of what we expect based on channels, based on the components that comprise this system.

It’s possible for a simulation to get down to very granular time resolution, down to picoseconds, tracking individual photons and tracking the quantum state as it evolves. In this way, we're able to validate the design.

How do you know your quantum network simulator is accurate?

Theoretical models already exist that can help in the design and validation of a quantum network, by using a formula to determine the raw key generation rate. However, theoretical mathematical models can become intractable very quickly as more nodes are added to the network. To go beyond those theoretical models, a quantum network simulation must be accurate and hold up in real-world scenarios.

One way to test the accuracy of a quantum network simulator is to compare the raw key rate vs. coincidence window for the theoretical formula, real world experiment, and the quantum network simulation. Below is a graph showing raw key rate vs. coincidence window in theory (blue dashed line), experiment on BrightNet (red markers), and simulation using AliroSimulator (gray markers).

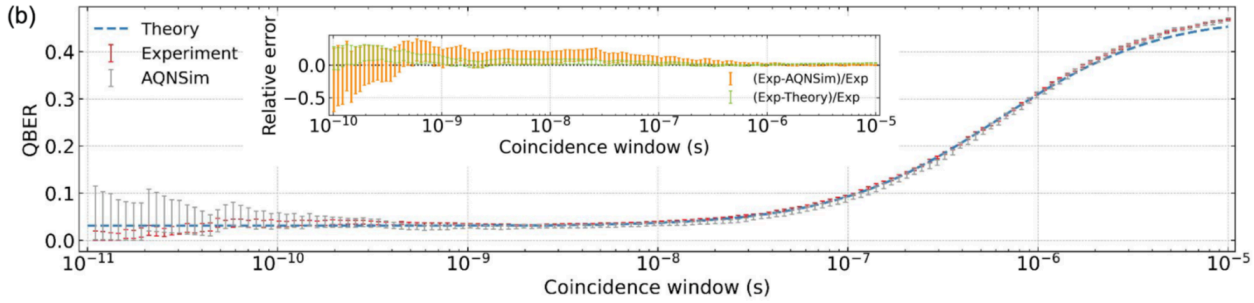


A coincidence window is the time bin used to identify clicks at end nodes as pairs of entangled photons. As the coincidence window widens, more true pairs are identified, so the raw key rate rises. Make the window too wide, and uncorrelated clicks (accidentals) will be counted, falsely inflating the raw rate.

The above graph (a) shows the trends we expect across theory, real-world physical behavior, and quantum network simulation. The inset of the graph shows relative error vs. coincidence window. The orange marks of the inset, representing simulation data, shows that the simulation’s relative error rate stays near zero as the coincidence window increases, with predictions within a few percent of the lab data. In contrast, the green marks of the inset, representing the predictions of the theoretical model, take a nose dive because the formula

overestimates the uncorrelated clicks (accidentals) leading to higher error rates as coincidence windows become larger.

Another test for the accuracy of a quantum network simulator is to compare the Quantum Bit Error Rate (QBER) vs. coincidence window using the theoretical formula, real world results, and the quantum network simulator. Below is graph (b) showing QBER vs. coincidence window in theory (blue dashed line), experiment (red markers), and simulation (gray markers).



As the distances between nodes increase, photon loss also increases in the theoretical model, the real world results, and the simulation. The theory, the physical network, and the quantum network simulation all follow the same trend, as expected.

These results show that a quantum network simulator can be used confidently to design and validate a quantum network, as a sanity check and also as a tool to optimize the network performance.

For details on the benchmarking discussed here, please see Realistic quantum network simulation for experimental BBM92 key distribution. https://arxiv.org/abs/2505.24851

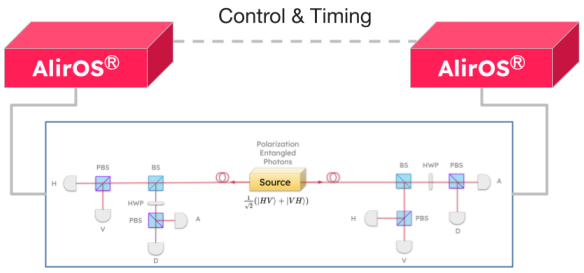
From Quantum Network Simulation to Quantum Network Deployment

Designing and validating the quantum network with an accurate and realistic quantum network simulation tool is the first step on the path to full-scale quantum network deployment. Through simulation, hardware components and protocols can be selected and constructed into a usable network, but quantum network simulation should be used throughout this process to optimize and plan for edge cases.

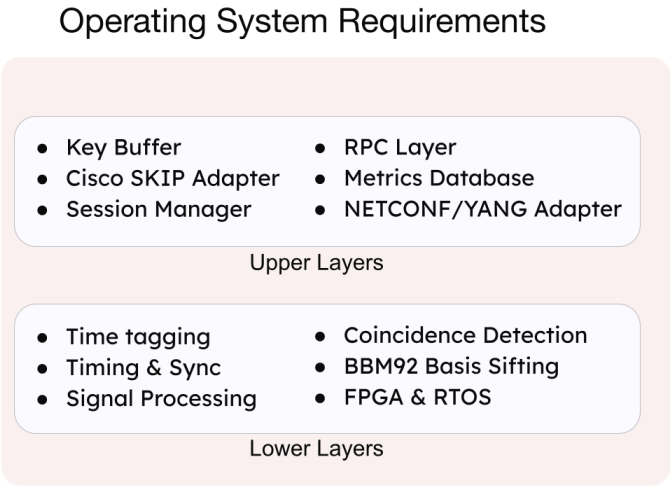
Designing a Link Layer to Support the Physical Layer

The physical layer of the network is made of quantum hardware components. While it can be somewhat unreliable and loss ridden on its own, it can be shaped into a reliable and robust system. On top of the physical layer of the network is the link layer, which is responsible for driving and monitoring the devices in the physical layer. There are many requirements and subsystems involved in the link layer, which is essentially a classical control layer that

coordinates the quantum physical processes. There are many supporting nested systems within the link layer, as pictured below.



- Nanosecond-level precision Timing & synchronization
- Real-time OS and FPGA logic for BBM92 KeyGen services
- Hardware abstraction layer (vendor-agnostic)
- Containerized upper layers



Quantum hardware can be difficult to set up, and in some cases quite expensive. Building and iterating solely on quantum hardware can be a slow and costly process. This process can be made much more economical in time and money spent by using a quantum network digital twin.

Up to this point, we have focused squarely on how simulation can be useful in the design, validation, optimization, and deployment of a quantum network, using Aliro's BrightNet network in Boston as a real-world example of this process. Using a digital twin moves the needle closer to emulation of the quantum network, making it possible to emulate the expected performance of the single-photon detectors.

Why emulate the single-photon detectors?

Once entangled photons reach a single-photon detector, the moment that photon is detected, the quantum state is measured and the detector outputs a classical electrical pulse: a time-stamped click. From that point on, everything is classical signal processing. The digital twin emulates this system by generating realistic streams of detector clicks with the same jitter, dead time, dark counts, and rates that would be present in the real-world components making up the physical layer. The digital twin emulates the quantum network and feeds the link layer this classical information.

There are many benefits to using a digital twin to design a link layer. No quantum hardware is required, making it more affordable than iterating with real components. For example, photon loss is a very common source of errors in a quantum system, but there are other errors that

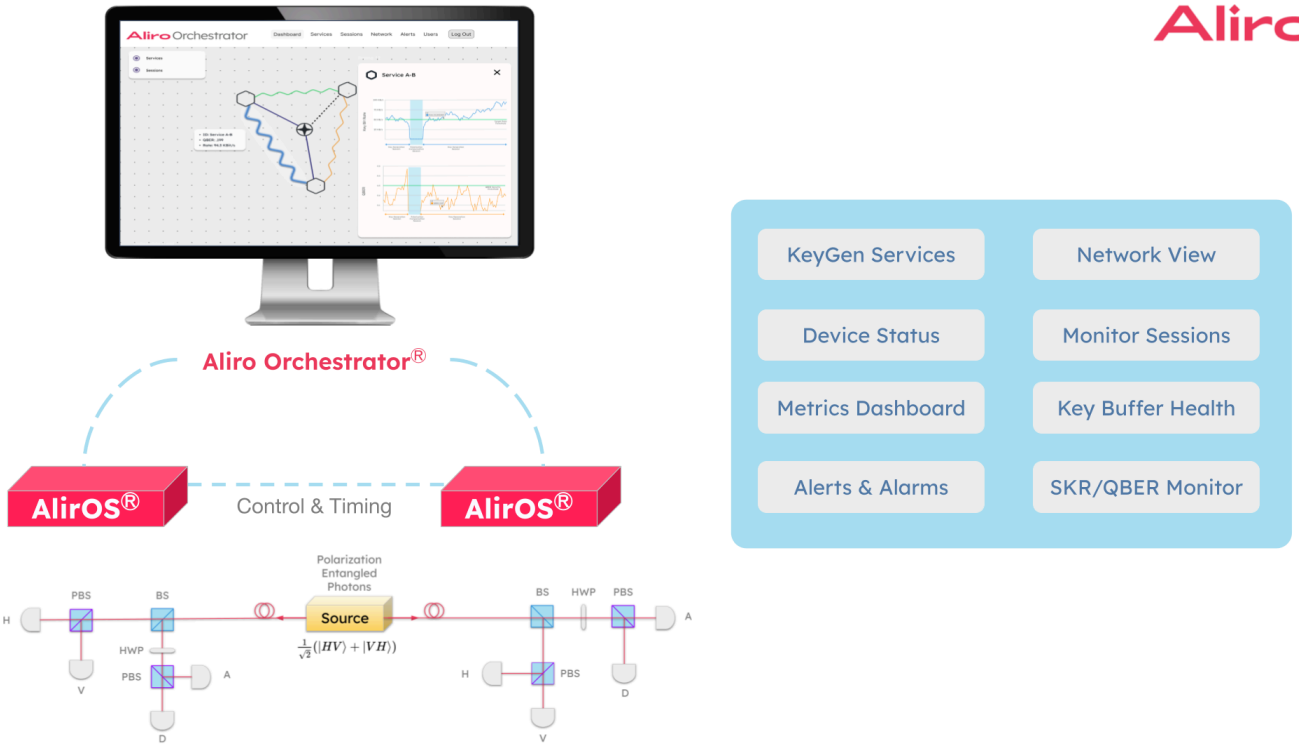
are less common or hard to reproduce on-demand. For example: what happens when clocks go out of sync? What happens if a message between Alice and Bob gets lost? What if polarization drift in the fiber becomes so large that Alice and Bob no longer see the expected correlations? These are scenarios and edge cases that must be carefully considered for a quantum network deployment in a real-world setting, such as a commercial deployment or a defense deployment. Testing these scenarios in a real quantum network might actually damage some quantum hardware, or potentially just very hard to produce the failure scenarios that the link layer needs to cover. Using a digital twin to emulate the quantum network provides a purely classical sandbox, one that is safe, relatively fast, and inexpensive compared to iterating with real quantum hardware. Emulation significantly de-risks the process of quantum network integration.

Managing the Quantum Network

Just like the classical internet, once constructed, the components of a quantum network need to be repeatedly configured, calibrated, monitored, updated, and orchestrated.

Quantum network orchestration runs the quantum network and makes it a usable service where network operators can do things like:

- Define who can request keys.
- Set max key age.
- See device health at each node (sources, detectors, clocks).
- View live metrics such as secret key rate, QBER, and loss.
- Track key buffer capacity and consumption.
- Set alerts & alarms for a range of parameters.
- View the topology with a graphical interface.

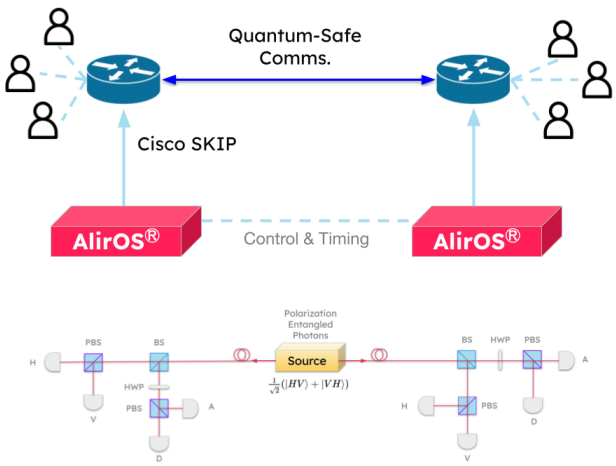


<SLIDE 19>

BBM92 in Action: Integrating BrightNet with Cisco SKIP

For a quantum network to provide Quantum Secure Communications, it will need to integrate with the encryptors and decryptors that are already deployed across the world. Aliro integrated BrightNet with Cisco encryptors and decryptors using the Cisco SKIP protocol.

The Cisco SKIP protocol is the interface that BrightNet uses to deliver the quantum generated keys that are stored in the key buffer. Once quantum keys are delivered, a quantum-safe communications link is established.
There are many ways to integrate quantum keys and different layers of encryption that can make use of them. In this case, Aliro implemented a quantum-safe VPN with a configurable key refresh rate, integrated with IPsec, which is a security protocol that uses AES-256 encryption.



- ✓ Full-stack live network
- ✓ IPsec VPN
- ✓ AES-256 encryption
- ✓ Cisco SKIP interface
- ✓ Configurable key refresh rate
- ✓ Quantum-safe data transfer using BBM92 keys

Each side of the network diagram above represents an end node. At the bottom, you can see the entangled photon source at the center of the physical layer. Above the physical layer is the link layer. Using the Cisco SKIP protocol, the link layer sends entanglement-generated keys to the classical network routers containing encryptors / decryptors that use the keys to secure communications.

To see this in action, please watch our on-demand webinar *Quantum Networks in Action: How BBM92 is Delivering Future-Proof Security.* https://www.brighttalk.com/webcast/19861/644059

A Full-stack Solution for Quantum Networking

Entanglement-based secure networks are being built today by a variety of organizations for a variety of use cases, benefiting organizations internally as well as providing great value to an organization's customers.

Telecommunications companies, national research labs, intelligence organizations, and systems integrators are just a few examples of the organizations Aliro is helping to leverage quantum networking.
Building entanglement-based quantum networks is no easy task. It requires:

- Emerging hardware components necessary to build the quantum network.
- The software necessary to design, simulate, run, and manage the quantum network.
- A team with expertise in quantum physics and classical networking.
- Years of hard work and development.

This may seem overwhelming, but Aliro is uniquely positioned to help you build your quantum network. The steps you can take to ensure your organization is meeting the challenges and

leveraging the benefits of the quantum revolution are part of a clear, unified solution already at work in quantum networks like the EPB Quantum Network℠ in Chattanooga, Tennessee.

AliroNet™, the world's first full-stack entanglement-based quantum network solution, consists of the software and services necessary to ensure customers will fully meet their quantum networking goals. Each component within AliroNet™ is built from the ground up to be compatible and optimal with quantum networks of any scale and architecture.

AliroNet™ is used to simulate, design, run, and manage quantum networks as well as test, verify, and optimize quantum hardware for network performance. AliroNet™ leverages the expertise of Aliro personnel in order to ensure that customers get the most value out of the software and their investment.

Depending on where customers are in their quantum networking journeys, AliroNet™ is available in three modes that create a clear path toward building full-scale entanglement-based secure networks: (1) Emulation Mode, for emulating, designing, and validating quantum networks, (2) Pilot Mode for implementing a small-scale quantum network testbed, and (3) Deployment Mode for scaling quantum networks and integrating end-to-end applications.

AliroNet™ has been developed by a team of world-class experts in quantum physics and classical networking. To get started (or continue on your quantum journey), reach out to the Aliro Quantum team for additional information on how AliroNet™ can enable your quantum network.

info@alirotech.com www.alirotech.com