Received: 9 February 2021 | Revised: 17 July 2021 | Accepted: 4 August 2021

IET Quantum Communication

DOI: 10.1049/qtc2.12019

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

# Timing and synchronisation for high-loss free-space quantum communication with Hybrid de Bruijn Codes

Peide Zhang[1] | Daniel K. L. Oi[2] | David Lowndes[1] | John G. Rarity[1]

[1]Department of Engineering, University of Bristol, Bristol, UK

[2]SUPA Department of Physics, University of Strathclyde, Glasgow, UK

**Correspondence**

Peide Zhang, Department of Engineering, University of Bristol, BS8 1UB, Bristol, UK.
Email: peide.zhang@bristol.ac.uk

**Funding information**

China Scholarship Council, Grant/Award Number: 201906840125; Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/T001011/1, EP/T517288/1; University of Bristol, Grant/Award Number: 1957333; UK Space Agency, Grant/Award Number: NSTP3-FT2-065

**Abstract**

Satellite-based, long-distance free-space quantum key distribution has the potential to realise global quantum secure communication networks. Detecting faint quantum optical pulses sent from space requires highly accurate and robust classical timing systems to pick out signals from the noise and allow for reconciliation of sent and received key bits. For such high-loss applications, a fault-tolerant synchronisation signal coding and decoding scheme based on de Bruijn sequences is proposed. A representative synchronisation timing system was tested in laboratory conditions and it demonstrated high fault tolerance for the error-correction algorithm even under high loss. The performance limitations of this solution are also discussed, and the maximum error tolerance of the scheme and the estimated computational overhead are analysed, allowing for the possibility of implementation on a real-time system-on-chip. This solution not only can be used for synchronisation of high-loss channels such as channels between satellites and ground stations but can also be extended to applications with low loss, high bit error rate, but require reliable synchronisation such as quantum and non-quantum communications over terrestrial free space or fibre optic channels.

**KEYWORDS**

cryptography protocols, photons, quantum communication, quantum cryptography, quantum information

## 1 | INTRODUCTION

Quantum Key Distribution (QKD) is able to secure communication links making them immune to quantum computer-based attacks that threaten the existing deployed public key cryptosystems such as the Rivest–Shamir–Adelman (RSA) algorithm and Elliptic Curve Cryptography (ECC) [1]. Current commercial QKD systems are deployed over optical fibre but intrinsic exponential losses restrict their range to under 1000 km, typically tens to hundreds of kilometres in practice [2, 3]. Satellite QKD has been proposed as an alternative to establishing intercontinental secure communication links, with the pioneering Micius satellite providing in-orbit-demonstrations of the concept [4, 5]. However, transmitting faint quantum optical pulses between a satellite and the Earth is challenging due to high channel losses and rapid relative motion between the transmitter and receiver. In order to pick out the low rate of detection events against background noise, classical laser beacons provide precise synchronisation so as to reject the detection events outside of the expected time of arrival of each quantum signal pulse. Additionally, each received detection event must be identified with its transmission pulse; hence, there also needs to be a mechanism to establish absolute pulse slot numbering.

There has been a general move towards space systems' miniaturisation with the rapid take-up of small satellite systems such as CubeSats (1–10 kg) [6, 7]. This facilitates the deployment of QKD satellite constellations to provide widespread and low latency global coverage that would be difficult and/or expensive to achieve with conventionally sized satellites (Micius is ~650kg). But the constraints on size, weight, and power (SWaP) of CubeSats pose additional challenges for the design and operation of the auxiliary communication systems required for QKD operation [8]. It is also desirable to reduce

the cost and complexity of the ground segment, that is, optical ground station (OGS), so they may be deployed in large numbers, some of which may be mobile or transportable [9].

## 1.1 | Background

In a free-space QKD communication system, just like for most telecommunication systems, Alice and Bob require a way of synchronising the transmitted and received signals. This may be achieved through synchronising clocks at either end, and referencing the signals to these local oscillators together with the time-of-flight delay. This allows the receiver measurements to be put in correspondence with the transmitted signals. Improving synchronisation also allows smaller coincidence windows to be used, hence increasing signal to noise ratio and lowering the quantum bit error rate. Therefore, it is essential to establish an efficient and reliable timing and synchronisation (T&S) mechanism.

For example, for free-space optical communications, Khader et al. used binary-phase-modulated continuous-wave laser to achieve better than 1 ps error between clocks situated 4 km apart over several hours, despite high system complexity and cost [10]. For quantum positioning, since the accuracy of time synchronisation directly determines the accuracy of positioning, a series of synchronisation methods between satellites and ground stations have been proposed, which could achieve a precision ranging from a nanosecond to sub-picosecond [11]. However, these methods are either based on constellations or coherent detection, requiring enough space or auxiliary facilities. Agnesi et al. used a laser ranging system to launch a 55 ps laser pulse to a satellite at a distance of 7500 km [12] achieving a detection accuracy of 230 ps of the reflected photons. But this method is based on the co-location of the transmitter and receiver and an internal clock network to gate the detection signal as well as a known target distance.

Several different T&S approaches have been utilised specifically for QKD over long free-space channels. Bourgoin et al. demonstrated QKD over simulated high-loss optical channels and full post-processing, using minimal computing hardware at the receiver [13]. They used a predefined random sequence for timing analysis, with the disadvantages of high storage requirements (storing the predefined sequence) and the need for fast processing. Similarly, the quantum satellite Micius used a combination of Global Positioning System (GPS) Pulse Per Second (PPS) and unencoded laser pulses for T&S, but at the cost of system complexity [14, 15]. More importantly, the above beacon pulses do not carry any absolute time information, so the raw secure key of Alice and Bob needs to be matched by a time-consuming correlation algorithm that increases the post-processing overhead. The Japanese mission SOTA demonstrated space to ground transmission of single-photon level polarised states [16]. It used a pseudorandom binary sequence (PRBS) to encode the signal pulse. Although the beacon carries absolute time information, recovery after extended fading may not be rapid. Additionally, the correlation-based matching algorithm is not very efficient. While the above techniques have been proven to work in their respective implementations, there has not been much research focussing on the requirements of resilience and speed utilising low resources.

## 1.2 | Contribution of this paper

Here, we present an alternative T&S method using a beacon with on-off modulation encoding a de Bruijn sequence. The beacon co-propagates with the quantum signals and provides both synchronisation information for identifying detection coincidence windows and absolute pulse positioning for QKD reconciliation. The de Bruijn sequence is generated deterministically in real-time, reducing storage requirements, and is optimal for locating the position in a bit stream as only an n-bit sub-string is needed to uniquely identify its location within a $2^n$ length cycle. The rising edges of the pulses are used to provide tight timing information, so that the arrival time windows of the quantum signal can be identified. The system displays good robustness as well as encoding and decoding efficiency.

The beacon modulation has also been designed to balance sequence encoding efficiency with timing jitter performance. For each bit in the de Bruijn sequence, we use two pulse slots to represent the binary code, on–on is 1 and on–off is 0. We call this a Hybrid de Bruijn Code (HDBC) and it avoids long periods with no pulses (e.g. for a long string of conventionally encoded 0 s) that would impact upon timing jitter. The first on-pulse of every encoded bit also allows monitoring of the pulse-loss ratio and link status.

Our de Bruijn-based T&S system has the advantage of rapid re-establishment of absolute pulse position after link loss, for example, due to turbulence-induced channel fade, requiring a minimum of received encoded beacon bits, allowing real-time reconciliation between the OGS and satellite. More importantly, the implementation of the synchronisation protocol can be directly deployed on the existing beacon system used for satellite acquisition, pointing and tracking (APT) without adding additional hardware, which is critical for a resource-limited CubeSat. Here, we experimentally test the resilience of the scheme to beacon corruption and devise schemes for error detection and correction of pulse position that can be straightforwardly implemented on embedded hardware.

# 2 | EXPERIMENTAL SETUP AND METHOD

We have developed a laboratory demonstration of an optical T&S link designed for pulse gating and numbering of quantum signals in satellite QKD (Figure 1). The transmitter employs a field-programmable gate array (FPGA) to generate a modulated pulse stream for the beacon signal that is sent to the beacon receiver consisting of a high-sensitivity avalanche photo-diode (APD) detector operating in a linear (non-Geiger) mode. For laboratory tests, we use an ND Filter to simulate beacon channel loss in the range of 50–70 dB. For testing, a

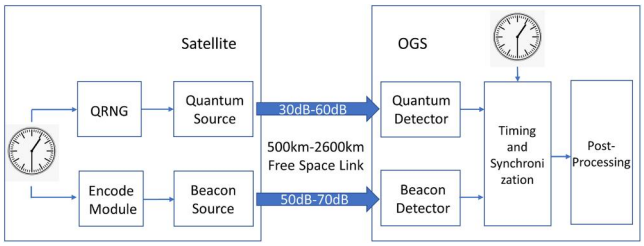


**FIGURE 1** High-level satellite Quantum Key Distribution T&S schematic. A beacon channel is established between the transmitter and the receiver allowing the optical ground station (OGS) to gate and synchronise the quantum signals sent by the satellite. The beacon produces high peak power, short duration optical pulses at a low duty-cycle. The quantum source is driven by a quantum random number generator (QRNG) to ensure cryptographic security of the distributed key (a pseudorandom binary sequence was used for testing). During an overpass, the channel losses vary from 20 to 30 dB due to changing range and atmospheric attenuation. The satellite to ground (downlink) beacon also is used to provide tracking and polarisation reference frame information for the OGS (not shown)

PRBS module is used to provide simulated electronics pulses to drive a quantum source that experiences 30–60 dB of channel loss. The quantum source and the beacon are synchronised by a shared internal clock network. In the receiver, the detector outputs of beacon and quantum channels are time tagged and processed by the T&S system in order to identify the quantum signal events and identify the pulse slot number index. Finally, through authenticated classical communication between the satellite and the OGS, the received quantum signals would be post-processed to generate a final secret key string; this final step was omitted in the demonstration of the T&S system.

In combinatorial mathematics, a de Bruijn sequence of order $n$ on a size-$k$ alphabet $A$ is a cyclic sequence in which every possible length-$n$ string on $A$ occurs exactly once as a sub-string (i.e. as a contiguous sub-sequence).[1] Such a sequence is denoted by $B(k, n)$ and has length $k^n$, which is also the number of distinct strings of length $n$ on $A$. The de Bruijn sequence has been widely used in various fields [18]. Due to the de Bruijn sequence's high efficiency, it has become an important navigation method in automated guided vehicles. For example, by placing a series of de Bruijn-coded binary colour pattern on a robot's route, it can decode its absolute position by scanning the local pattern. This solution eliminates the need for high-precision sensors and reduces costs, which is especially important in industrial applications [19, 20]. Howie et al. used de Bruijn-coded long exposure photographs to detect meteor fireballs and efficiently decode the meteorite trajectory from the picture [21]. In addition, de Bruijn sequences and their associated graphs can be constructed for grid network topologies [22] used in distributed hash table protocols [23], gene tag coding in bioinformatics [24], and other applications [25].

[1] As with most "named" results, de Bruijn was not the first to discover such sequences [17].

In Figure 2, an HDBC pattern example $B(2, 4)$ is given. Here, we use the BB84 QKD protocol [26] as the example. The four transmitted polarisation quantum signal pulses, H, V, A, D are all synchronised with the beacon channel. Reliably obtaining the arrival time and index of each quantum pulse at the receiver is critical to the post-processing steps of the protocol. The clock and the red pulses (without binary label) are not modulated, thus allowing them to be used for precise timing on the rising edge. To improve the suppression of noise signals, the pulse peak power and rise time should be chosen carefully. The blue pulses (with binary label) are encoded in the de Bruijn sequence, which delivers the absolute pulse index of each quantum detection to the receiver. For high-loss applications, increased peak pulse power is required to maintain low timing jitter. To keep the average beacon power as low as possible, the repetition rate and duty cycle of the beacon is reduced compared with the quantum signal. The quantum key rate is kept high (typically 100 MHz); thus, the recovered beacon clock (at 100 kHz) needs to be interpolated before synchronising with the quantum signal. After interpolation, the clock pulses not only can supply a very stable reference edge but also give us a beacon error rate (ratio of undetected to detected pulses) in the case of high channel loss. The de Bruijn pulses determine the position numbering of the quantum signals from the start of the transmission with extremely high decoding efficiency.

To minimise the memory and computational requirements for generating de Bruijn sequences, a linear-feedback shift register (LFSR) method is used. An LFSR is a shift register whose input bit is a linear function of its previous state. The most used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value. Due to these two features, the structure is very suitable for implementation in an FPGA system [27]. On the one hand, this structure only needs to be implemented with a few of the logic elements, which means the logic resource requirements for FPGA are particularly low; on the other hand, each clock cycle can generate a new bit which brings the possibility of real-time coding removing the need to store a predefined sequence in memory.

The HDBC sequence encode and decode scheme is shown in Figure 3. The left block is the transmitter on the satellite which includes a code generator and a laser pulse driver module. For the link between satellite and OGS, the beacon channel loss for our CubeSat-based QKD system is in the range of 50–70 dB. More importantly, it will suffer from noise including turbulence, background light, fluctuating cloud cover, all of which may even cause a link interruption (or fade); hence, a reliable code pattern that can correct errors is critical. The right block on the diagram is the decoding workflow. After being received by the time tagger, the pulse train is first decomposed into two sub-strings, half is the de Bruijn sequence and the other half is the clock signal. The clock sequence should have a near constant period without any modulation, other than that due to Doppler chirp caused by satellite motion. Inputting this to the error-ratio monitoring

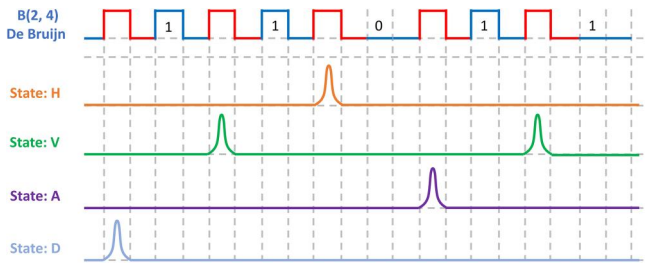


**FIGURE 2** Hybrid de Bruijn Code (HDBC) pattern. A HDBC pattern is designed for the purpose of timing and synchronisation. It interleaves unencoded clock pulses with a de Bruijn sequence, so that it can carry both the position and time information of the quantum signal. The quantum signals (H, V, A, D) are detected within the arrival time slots that are calculated by the receiver from the beacon timing information carried by the rising edges. Noise counts detected outside the pulse slots are rejected, and the degree of suppression is given by the ratio of the pulse slot width and the repetition period of the quantum signal. The pulse slot index number is determined from decoding the de Bruijn encoded sequence. In practice, the beacon signal rate is much lower than that of the quantum signal transmission, hence the pulse slot times and indices are interpolated between beacon pulses

module allows an estimate of the fraction of lost pulses that serves as an estimate of the error ratio of the de Bruijn channel. Since the errors are almost entirely caused by the loss of pulses, the error distribution is asymmetrical, that is it is more likely that a '1' becomes '0' than vice versa.

Once a sub-sequence of beacon pulses is received, the absolute position within the entire sequence must be determined. Traditional binary string matching needs to perform correlations over large subsets of bits stepping through many delay settings, which leads to complex and slow matching algorithms. According to the property of de Bruijn sequence, every possible length-$n$ string in the sequence occurs exactly once as a sub-string, so we use a look-up table matching method (LUTM) to greatly improve the matching efficiency (see Figure 4). We precompute a table to store the position index of all sub-string patterns that can be efficiently searched once a sub-string has been received.

The absolute pulse index decoding procedure needs to be robust against errors in the received beacon pulses. Since Hamming invented the first error correction code (ECC) in 1950 (the Hamming (7, 4) code), it has been widely used in computing, telecommunication, information theory, and coding theory [29]. The central idea is that the sender encodes the message with redundant information included within the ECC that allows the receiver to detect a limited number of errors that may occur anywhere in the message, and often correct these errors without re-transmission. This means that the amount of data actually sent is greater than the actual information to be communicated. There are many kinds of ECCs for different circumstances, for example, the AN code, BCH code, Walsh–Hadamard code and so on [30–32]. However, for the de Bruijn code, due to its special properties, a sub-string with a length greater than $n$ contains additional information that allows the detection of errors without transmitting redundant data. In an ideal de Bruijn sub-sequence, the position indices of adjacent pulse sub-strings should continuously increase by one. However, the presence of errors leads to the decoded position index advancing discontinuously, or even if it is still continuous, all positions are offset by a value. These are the two situations the decoding algorithm must overcome in an error-corrupted de Bruijn sequence:

**Error pattern A:** The decoding results of adjacent pulses have jumps and do not show a smooth increment.
**Error pattern B:** The decoding results of adjacent pulses still show a smooth increment, but they are offset by a value from the original position.

As shown in Figure 5a, in the first case, the error bit can be easily detected by finding the break point where the position index is not advancing uniformly. However, if only the continuity of the decoding result is used to judge whether the decoding is correct, then we will mistakenly regard the second case to be correct. To solve this problem, we need to increase the sub-sequence length $L_s$ for decoding, but a longer sub-sequence introduces additional computational overhead. Hence, we are interested in how short a sub-sequence can be and still be able to reliably detect and correct errors. From another perspective, considering the random distribution of errors, the question is what error rate the decoding algorithm can tolerate.

There is the possibility that some sub-sequences include more than one error bit, exceeding the upper error tolerance limit, therefore an improved decoding strategy is needed. First, the algorithm finds a $2n$ sub-sequence without any error bits. This step is significant and complex because we need to guarantee its correctness. As Figure 5b shows, the algorithm first finds a $2n$ sub-sequence with a continuous position index, then it moves forward to the next $2n$ sub-sequence with continuous position indexes and no overlap. Then, if the time offset between two sub-sequences is equal to the difference in position index between the two sub-sequence after decoding, we can be confident that the first $2n$ sub-sequence is correct. Once the algorithm has found the first corrected sub-sequence (frame header), each time it moves forward one bit, it should find the next sub-sequence. Even if there is an error bit, it will appear on the last bit. So, after the frame header is found, each time the algorithm just needs to pick an $n+1$ sub-sequence, then two pulses positions can be decoded. If they are continuously increasing, then the sub-sequence is correct, otherwise, the last bit is wrong. The improved decoding process can be summarised as follows:

Step 1: Find the first correct sub-sequence (frame header).
Step 2: Each time pick a length-$(n+1)$ sub-string, decode two position indexes. If the two values are continuously increasing, then move forward 1 bit and repeat Step 2. If not, jump to Step 3.
Step 3: Detect the position of the error bit, correct it and return the right result. Update the error-ratio

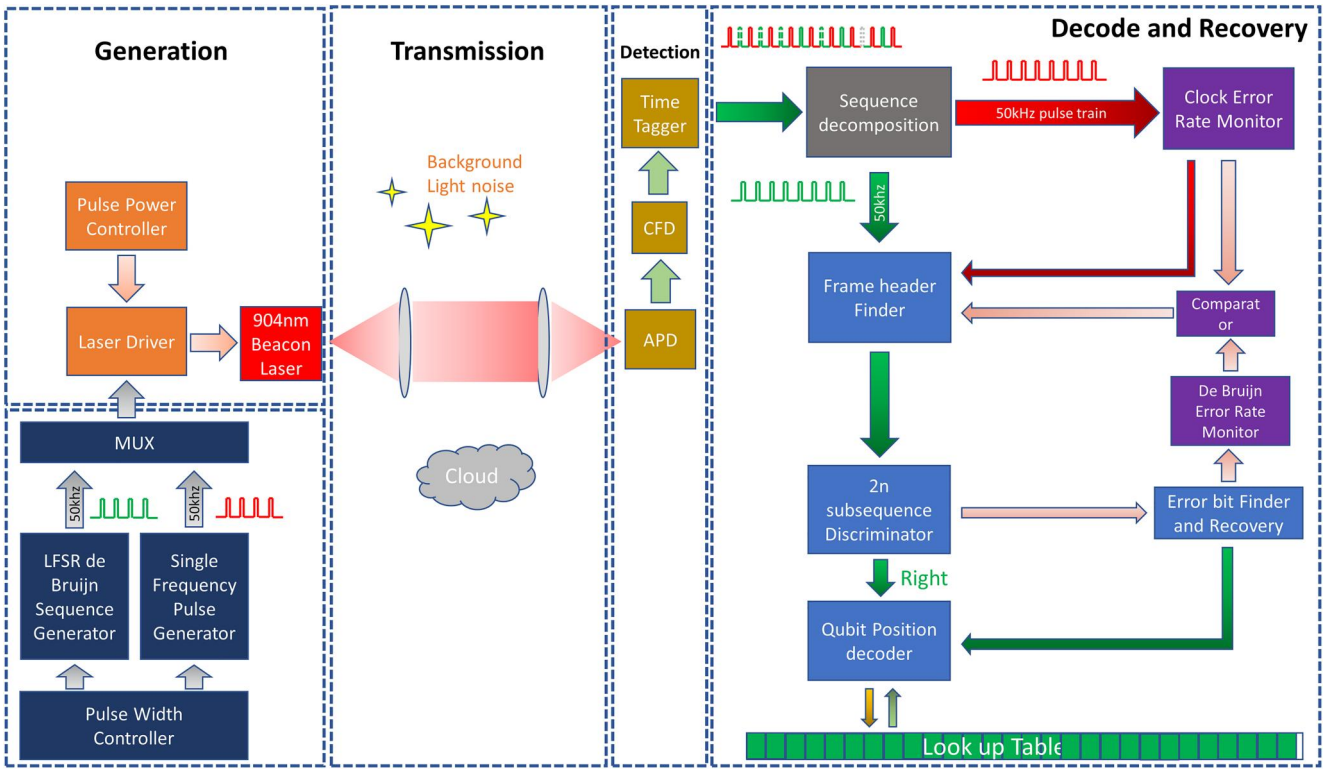


**FIGURE 3** De Bruijn sequence encode and decode diagram. There are four main steps through the process. The first step, generation, creates a beacon laser signal encoded using the Hybrid de Bruijn Code (HDBC). A Xilinx Spartan 6 FPGA is used for electronic pulse generation and modulation. Within the FPGA, a pulse width controller sets the pulse duration of the beacon. A linear-feedback shift register (LFSR) de Bruijn sequence generator can generate a de Bruijn sequence with a repetition of 50 KHz in our configuration. A single-frequency pulse generator is used for generating an unencoded pulse as the clock, the MUX is used for interleaving the two separate sequences to produce a 100 kHz HDBC sequence. A laser driver board was designed in the lab to power a laser diode (Thorlabs LP904-SF3) with a wavelength of 904 nm. A pulse power controller adjusts the optical pulse peak power and the laser driver converts the electronic pulse into the laser pulse with the configured parameters. The transmission stage mainly describes the signal channel characteristics such as background light noise, Cloud, turbulence and so on. All of these factors will affect the jitter between the beacon and quantum signal and even introduce errors into the sequence pattern. In the Detection block, the optical pulse is converted into an electronic pulse which is sampled. The APD (Hamamatsu S13282) has a very high conversion gain of ~3.4MV/W at 904 nm. A constant fraction discriminator (CFD) is used to reduce signal jitter/walk due to signal amplitude variations. A Time Tagger [28] is used for recording the absolute time of the arrival pulse. The final step is decode and recovery which is implemented by a software algorithm. Sequence decomposition separates the de Bruijn sequence and clock. The frame header finder is used to find an acceptable location and start decoding from that location. The $2n$ sub-sequence discriminator is used to determine whether the sequence contains error bits. The Qubit position decoder will decode the pulse position based on the pre-defined look-up table when the sub-sequence is correct. The error bit finder and recovery module can find the error bit position in the sub-sequence and correct it. The clock error rate monitor and the de Bruijn error rate monitor tracks the error rates of the two sub-sequences in real time, and the comparator will trigger the decoding interrupt when the two error rates have a significant difference

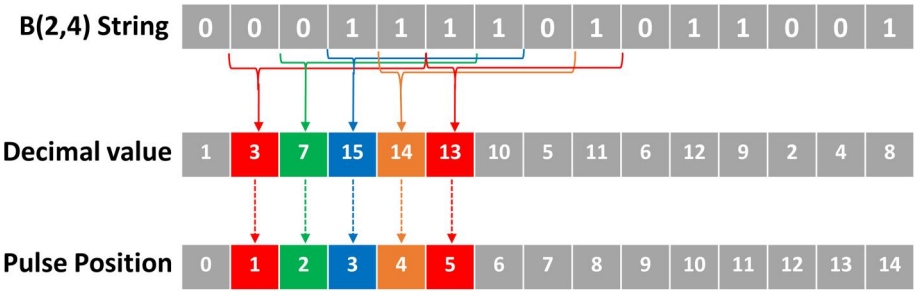


**FIGURE 4** Look-up table matching method (LUTM) structure. The binary value of each sub-string is converted to decimal and is stored alongside its position in the sequence. The sorted table is pre-generated before the communication. This greatly improves decoding efficiency at the expense of a small amount of storage space

record, if the error ratio differs greatly from the clock error-ratio, then jump back to Step 1; if not, jump back to Step 2.

The optical link between the satellite and OGS may be interrupted due to cloud cover, pointing disturbance, or a burst of turbulence leading to an extended loss of pulses. When the

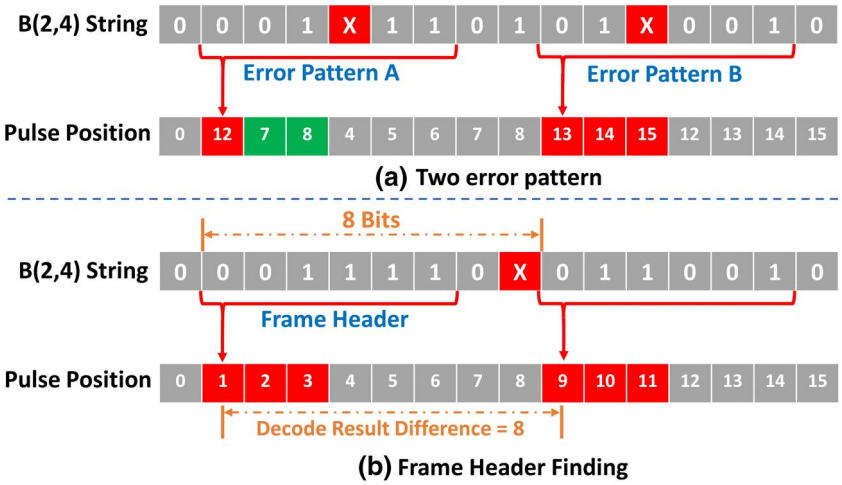


**FIGURE 5** Error detection and correction scheme. (a) Two error patterns of the de Bruijn sequence. In the first situation, the decode result does not increase steadily due to an error; in the second situation, the decode result is successive, but the results have an offset compared with the actual position. (b) Frame header finding method

link resumes, we would like to re-establish index decoding as quickly as possible with the minimum number of beacon bits. The clock pulses are used for monitoring the link and if the OGS fails to detect them, it means the link has been interrupted. The clock pulses can also be used for monitoring the pulse loss ratio in real time. The two signals passing through the same channel should have a similar loss; if the pulse loss ratio of the clock and de Bruijn pulses differ significantly, it means the decoding algorithm has experienced a problem and the decoding process will stop and restart. Each time when there is an interruption, the decoding process needs to restart, and we need to re-find an uncorrupted frame header.

## 3 | RESULTS

For the system introduced above, we experimentally tested its resilience against noise and interrupt recovery ability. An artificial noise source is used to add errors to the ideal signal pulse. The noise source digitally generates a Gaussian distributed random signal, and the required noise level can be set by adjusting the threshold. The effect of the noise is characterised by the bit error rate (BER). Also, we define another factor called the decoding bit error rate (DBER) to describe the performance of the decoding process. Although the de Bruijn code for timing may be recovered by more intensive post-processing of detector time tags, the real-time decoding accuracy is still an important evaluation criterion for the performance of a deployed, low-processing footprint ground terminal. A higher decoding accuracy reduces the need for additional computation, memory, and time on post-processing, and most importantly makes real-time reconciliation a possibility.

A performance comparison between decoding with or without the error correction is shown in Figure 6 where a significant improvement can be seen due to the correction algorithm, especially at large values of the BER. Below a BER of 0.2, the DBER remains low, not exceeding 0.03. Figure 7 shows in detail the performance of the error correction algorithm. Figure 8 shows the results of a complete encoding and decoding experiment.

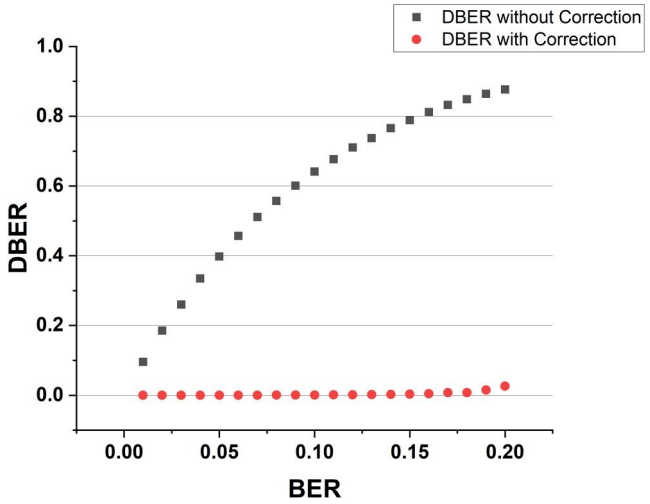


**FIGURE 6** Decoding error versus bit error rate (BER) for $B(2, 20)$ [33]. The dark boxes represent the decoding bit error rate (DBER) without error correction, from which we could see that the DBER increases logarithmically with the increase of BER and eventually tends to 1 for high BER. The red circles represent the DBER with error correction. Compared with the dark boxes, the DBER does not increase significantly between BER values of 0% to 20%, and it always remains lower than 3%

## 4 | PERFORMANCE ENVELOPE

For a timing and synchronisation method applied in high-loss situations, it is critical to evaluate the error tolerance and computational complexity of the algorithm. In this section, we discuss the maximum error rate the algorithm can tolerate giving an indication of the harshest environment in which this solution can be used. In addition, we discuss the computational overhead of this solution pointing the way to real-time implementation of the decoding scheme on a dedicated system-on-chip.

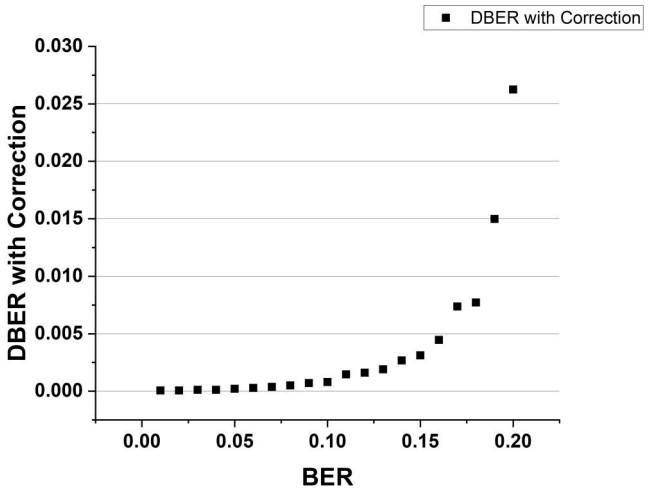


**FIGURE 7** Decoding bit error rate (DBER) versusbit error rate (BER) with error correction for $B(2, 20)$ [33]. This shows a zoomed portion of Figure 2 indicating the DBER for even moderate to high BER. Performance degrades gracefully up to $BER \sim 0.15$, above which the decoding accuracy deteriorates quickly

## 4.1 | Maximum error rate that can be tolerated

First, we will still use $BER$ to evaluate the error rate in the raw data. Specifically here, we can explore how short a sub-string in which an error bit can be detected. According to the characteristics of a de Bruijn sequence $B(2, n)$, each sub-sequence of length-$n$ can be decoded to obtain a corresponding position index within a cycle of length $2^n$. If we want to get the position information of $m$ adjacent pulses, we need to pick a sub-sequence of length $n + m - 1$. Let us assume that there is, on average, a single bit error in this sub-sequence; hence, the $BER$ is

$$BER = \frac{1}{n + m - 1}, \ m > 0. \tag{1}$$

Here let us consider two cases, one is when $0 < m \leq n$ and another case is when $m > n$. In the first case, the worst situation is that the error bit happens to be in the middle of the sub-sequence, which means each length-$n$ sub-sequence will include the error bit, and it results in the second wrong decoding situation, that is all the pulses' position indexes are still continuous but with an offset. In the second case, let us first consider a sub-sequence of length $2n$. Wherever the error bit is, there is always an $n$-bit sub-sequence that does not contain it. This means that even if all the other $n$-bit sub-sequences' decoding results vary continuously and have an offset relative to the original position, the error free sub-sequence will supply the correct position, and there must be a break in continuity between the correct and adjacent incorrect sequences. This rule gives us a basis for detecting error bits in sequence with high confidence. From a statistical point of view, this provides an approximate upper limit of $BER$ for reliable decoding using a $2n$ sub-sequence:

$$\max BER = \frac{1}{2n}, \ n > 2. \tag{2}$$

## 4.2 | Decoding algorithmic complexity

In order to further verify the utility of this solution, we need to estimate its computational load. We have shown above that the decoding is highly efficient, and there is potential for real-time processing using a system-on-chip. The whole process can be divided into two parts: searching for a frame header and decoding all the following pulses. We assume a general binary de Bruijn sequence of length-n $B(2, n)$ and analyse the decoding complexity of a single pulse sub-sequence position. This is determined by the complexity of a single LUTM operation measured in clock cycles. This is then defined as our unit operation, and all subsequent complexity calculations are based on this unit.

### 4.2.1 | Frame header finding

After the commencement of a communication connection, the first phase is to find a definitely correct sub-sequence so as to ensure that in the subsequent decoding, each sub-sequence contains at most one error bit. Here, a factor $c$ needs to be introduced, which determines on average how many attempts must be taken to find a correct sub-sequence that does not contain an error from an arbitrary position in a noisy sequence. Assuming that the error rate of the sequence is lower than the upper error limit, then every time, we need to extract a sub-sequence with a length of $2n$ to verify whether it is without errors. Because we need to find two correct sub-sequences in different positions and further compare the distance between the two sequences and the difference between the respective decoding results, this is equivalent to finding a frame header. Therefore, the total algorithm complexity is

$$f(n) = 2c \cdot (n + 1), \tag{3}$$

where the constant $c$ is evaluated in the following.

Considering the statistical distribution of the intervals between two adjacent random error bits, we denote the probability of an interval greater than $2n$ as $\beta$ that depends on the $BER$ in the sequence. In a sequence of length $\frac{2n}{\beta} + n$, a completely correct sub-sequence of length $2n$ can be found with high probability. Here, assuming that the noise model obeys the Gaussian distribution, the relationship between $\beta$ and $BER$ can be obtained by analysing a series of experimental data. Figure 9 shows that when we use $n = 20$, the probability that the interval between two adjacent error bits is greater than $2n$ varies with the bit error rate. The empirical formula obtained by fitting is shown in

$$\beta(n) = 0.9394 \times e^{-\frac{(BER + 0.1016)^2}{0.1195^2}}, \ n = 20, \tag{4}$$

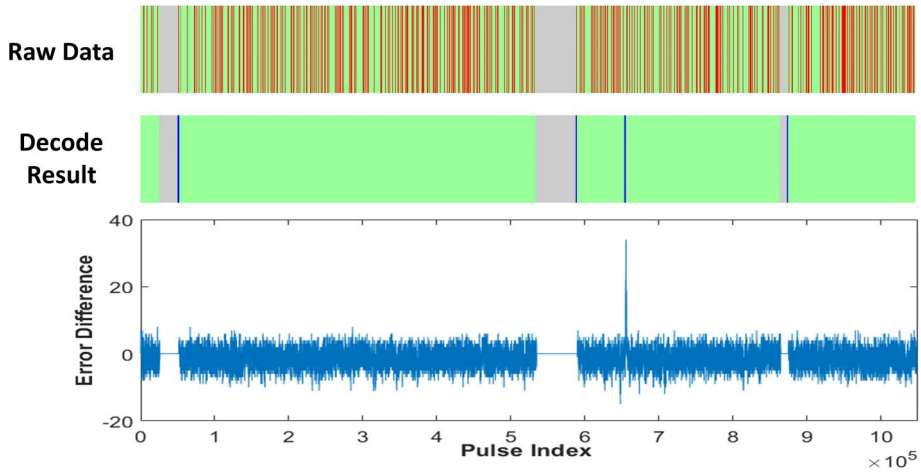


**FIGURE 8** Experimental decoding for $B(2, 20)$ for $BER = 0.15$ [33]. The first row shows the received pulse data with green representing correct bits, wrong bits in red, link loss by grey. In order to be clearer, the red error lines are displayed in bold. The second row shows the decode results, green for correct positions, blue for frame headers, and grey for the interrupted segments. We see that all the errors are detected and corrected, and every time when there is an interruption, the algorithm will restart to find a 'frame header'. The third row is the error rate difference estimated by counting the lost clock counts, and comparing with the errors in the predicted de Bruijn sequence, both accumulated in 1200 clock cycles. In the simulation, we artificially introduced an unknown decoding error at a random position indicated by the third blue line in the decode result, around pulse index $6.5 \times 10^5$. This error caused all the next decodes to show errors that would not be detected by the de Bruijn decode module. However, this potentially serious problem was detected by the error rate difference between the clock pulses and the de Bruijn decode module exceeding the threshold. This initiates a decode pause followed by a successful restart

The factor $c$

$$c = \frac{2n}{\beta(n)} + n, \tag{5}$$

and the total algorithm complexity is

$$O(frameheader) = O\left(\frac{4n^2}{\beta(n)} + 2n^2 + \frac{4n}{\beta(n)} + 2n\right). \tag{6}$$

The error interval ratio is related to the length of the selected substring, $\beta(n)$ in the formula is a function of $n$ thus changing for different lengths of de Bruijn codes. According to the upper limit of noise tolerance calculated in section A, the upper limit of the algorithm complexity is when the $BER = \frac{1}{2n}$.

In the situation where the timing and synchronisation is processed offline, the time required for its operation is less critical. However, for real-time operation, the delay in finding the frame header given by the above expressions is important because it represents the algorithm's response ability and recovery speed for link re-connection.

### 4.2.2 | Single pulse decoding

After finding the frame header, the algorithm will move forward one bit at a time on the main sequence, extract sub-sequences of length-$n + 1$, and decode them, so as to ensure that there is at most one error in each sub-string, and it must be at the lowest position. Ignoring the additional time needed for correcting detected errors, the algorithm complexity is

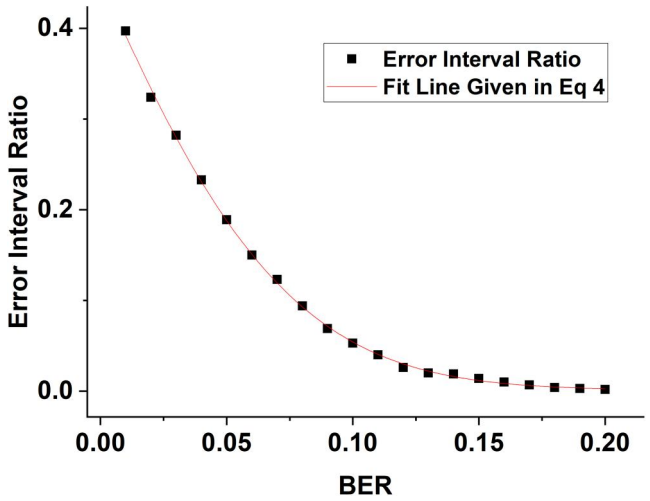


**FIGURE 9** The relationship between the error interval ratio ($\beta$) and *BER*

$$O(eachpulse) = O(2). \tag{7}$$

The algorithmic complexity of decoding each pulse is independent of the noise contained in the sequence, and the complexity is extremely low. In addition, since each LUTM operation does not include multiplication and does not involve subsequent data, the decoding operation of each pulse is independent of each other. These two characteristics make the algorithm very suitable for platforms such as FPGA and DSP to provide a basis for future real-time processing.

## 5 | CONCLUSION

In summary, we propose a synchronisation method based on de Bruijn sequences which is suitable for timing and synchronisation over high-loss communication channels. In space to ground quantum communications, the channel loss is particularly large, so it is particularly prone to beacon loss or even link interruption. An HDBC pattern has been designed to cope with these conditions so that errors in the timing sequence can be detected and corrected. In addition, we propose a sub-string matching method based on LUTM, which can greatly improve the matching efficiency and provide the possibility for real-time pulse reconciliation. Finally, we determine the maximum error ratio that can be tolerated, and study pulse pattern prediction based on the received pulse train.

The system has been designed to reduce the computational overhead required to establish the absolute pulse index, especially after extended link loss. The de Bruijn code provides an efficient sequence position encoding that has inherent error tolerance that is exploited in achieving robustness to beacon corruption in the decoding process. The rapid index recovery is especially valuable for real-time pulse reconciliation in (LEO) satellite QKD where the time for communication is restricted during the overpass. The proposed method has general applicability outside of satellite QKD. It is not restricted for use in the synchronisation systems of high-loss channels such as channels between satellites and ground stations but can also be extended to applications with high loss, high bit error rate, and the need for reliable synchronisation, for example, fibre-based quantum communication [34–36], other platforms like drones [37] and quantum secure direct communication [38, 39].

## ACKNOWLEDGEMENTS

This research study was funded by the EPSRC Quantum Communications Hub (EP/T001011/1), and the UK Space Agency (NSTP3-FT2-065 QSTP: Quantum Space Technology Payload, NSIP-N07 ROKS Discovery). The first author was supported by the University of Bristol–China Scholarship Council Joint-funded scholarship to participate in this research study. DO is supported by the EPSRC (EP/T517288/1) and acknowledges discussions with Craft Prospect Ltd., S. K. Joshi, M. Stefko and E. M. J. Hastings.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available at http://doi.org/[10.5523/bris.2n9e04k9mlpck2oy6mqp62bvs4] [33].

## ORCID

*Peide Zhang* https://orcid.org/0000-0003-4595-8151
*Daniel K. L. Oi* https://orcid.org/0000-0003-0965-9509
*David Lowndes* https://orcid.org/0000-0002-2364-4373
*John G. Rarity* https://orcid.org/0000-0002-8601-5558

## REFERENCES

1. Gheorghiu, V., Mosca, M.: Benchmarking the quantum cryptanalysis of symmetric, public-key and hash-based cryptographic schemes. arXiv preprint arXiv:1902.02332 (2019)
2. Ursin, R., et al.: Entanglement-based quantum communication over 144 km. Nat Phys. 3(6), 481–486 (2007). https://doi.org/10.1038/nphys629
3. Chen, J.-P., et al.: Sending-or-not-sending with independent lasers: Secure twin-field quantum key distribution over 509 km. Phys. Rev. Lett. 124, 70501 (2020). https://doi.org/10.1103/PhysRevLett.124.070501
4. Liao, S.-K., et al.: Satellite-relayed intercontinental quantum network. Phys. Rev. Lett. 120(1), 30501 (2018). https://doi.org/10.1103/PhysRevLett.120.030501
5. Bedington, R., Arrazola, J.M., Ling, A.: Progress in satellite quantum key distribution. NPJ Quant. Inf. 3, 8 (2017). https://doi.org/10.1038/s41534-017-0031-510.1038/s41534-017-0031-5
6. Mazzarella, L., et al.: Quarc: Quantum research cubesata constellation for quantum communication. Cryptography. 4(7), (2020). https://doi.org/10.3390/cryptography4010007
7. Sidhu, J., et al.: Advances in space quantum communications. IET Quat. Comm. 6 (2021)
8. Oi, D.K.L., et al.: Cubesat quantum communications mission. EPJ Quant. Technol. 4, 1–20 (2017)
9. Fuchs, C., et al.: DLR's transportable optical ground station. LTu1B–3 (2013)
10. Khader, I., et al.: Time synchronization over a free-space optical communication channel. Optica. 5, 1542–1548 (2018). https://doi.org/10.1364/OPTICA.5.001542
11. Duan, S., Cong, S., Song, Y.: A survey on quantum positioning system. Int. J. Model Simul. 41, 265–283 (2021). https://doi.org/10.1080/02286203.2020.1738035
12. Agnesi, C., et al.: Sub-ns timing accuracy for satellite quantum communications. JOSA B. 36, B59–B64 (2019)
13. Bourgoin, J.-P., et al.: Experimental quantum key distribution with simulated ground-to-satellite photon losses and processing limitations. Phys. Rev. 92, 11 (2015). https://doi.org/10.1103/physreva.92.052339
14. Wang, J.-Y., et al.: Direct and full-scale experimental verifications towards ground–satellite quantum key distribution. Nat. Photon. 7, 387–393 (2013). https://doi.org/10.1038/nphoton.2013.89
15. Yin, J., et al.: Satellite-based entanglement distribution over 1200 kilometers. Science. 356, 1140–1144 (2017)
16. Takenaka, H., et al.: Satellite-to-ground quantum-limited communication using a 50-kg-class microsatellite. Nat. Photon. 11, 502–508 (2017). https://doi.org/10.1038/nphoton.2017.107
17. Ralston, A.: De Bruijn sequences—a model example of the interaction of discrete mathematics and computer science. Math. Mag. 55, 131–143 (1982)
18. De Bruijn, N.G., Erdös, P.: On a combinatorial problem. Proc. Sect. Sci. Koninklijke Nederlandse Akad. Wetenschappen Amsterdam. 51, 1277–1279 (1948)
19. Berkowitz, R., Kopparty, S.: Robust Positioning Patterns, pp. 1937–1951. Society for Industrial and Applied Mathematics (2016)
20. Ortiz, J.: Absolute Position Measurement for Automated Guided Vehicles Using the Greedy de Bruijn Sequence (2006)
21. Howie, R.M., et al.: Submillisecond fireball timing using de Bruijn timecodes. Meteorit. Planet. Sci. 52, 1669–1682 (2017). https://doi.org/10.1111/maps.12878
22. Fraigniaud, P., Lazard, E.: Methods and problems of communication in usual networks. Discr. Appl. Math. 53, 79–133 (1994)
23. Kaashoek, M.F., Karger, D.R.: Koorde: A simple degree-optimal distributed hash table. In: International Workshop on Peer-to-Peer Systems, pp. 98–107. (2003)
24. Compeau, P.E.C., Pevzner, P.A., Tesler, G.: Why are de Bruijn graphs useful for genome assembly? Nat. Biotechnol. 29, 987–991 (2011)
25. Diaconis, P., Graham, R.: Magical mathematics: The mathematical ideas that animate great magic tricks. Princeton University Press (2015)
26. Bennett, C.H., Brassard, G.: Quantum cryptography. Theor. Comput. Sci. 175–179 (1984)

27. Chang, Z., et al.: On binary de Bruijn sequences from LFSRS with arbitrary characteristic polynomials. Des. Codes Cryptogr. 87, 1137–1160 (2018). https://doi.org/10.1007/s10623-018-0509-y
28. Nock, R., Dahnoun, N., Rarity, J.: Low cost timing interval analyzers for quantum key distribution. 2011 IEEE International Instrumentation and Measurement Technology Conference, Vol. 5 (2011). https://doi.org/10.1109/IMTC.2011.5944324
29. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Techn. J. 29, 147–160 (1950). https://doi.org/10.1002/j.1538-7305.1950.tb00463.x
30. Sun, Z., Zhu, S., Wang, L.: A class of constacyclic BCH codes. Cryptogr. Commun. 12 (2020). https://doi.org/10.1007/s12095-019-00401-6
31. Samanta, S., Maity, G., Mukhopadhyay, S.: All-optical Walsh-Hadamard code generation using MZI. 515–518 (2019). https://doi.org/10.1109/DEVIC.2019.8783836
32. Moreira, M., Guazzelli, R., Calazans, N.: Return-to-one protocol for reducing static power in C-elements of QDI circuits employing m-of-n codes. In: 2012 25th Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1–6. (2012). https://doi.org/10.1109/SBCCI.2012.6344444
33. Timing and synchronisation for high-loss free-space quantum communication with hybrid de Bruijn code. (2021). https://doi.org/10.5523/bris.2n9e04k9mlpck2oy6mqp62bvs4
34. Wengerowsky, S., et al.: Passively stable distribution of polarisation entanglement over 192 km of deployed optical fibre. NPJ Quant. Inf. 6, 1 (2020). https://doi.org/10.1038/s41534-019-0238-8
35. Joshi, S.K., et al.: A trusted node-free eight-user metropolitan quantum communication network. Sci. Adv. 6, eaba0959 (2020). https://doi.org/10.1126/sciadv.aba0959
36. Wengerowsky, S., et al.: Entanglement distribution over a 96-km-long submarine optical fiber. Proc. Natl. Acad. Sci. U. S. A. 116, 6684–6688 (2019). https://doi.org/10.1073/pnas.1818752116
37. Quintana, C., et al.: Low size, weight and power quantum key distribution system for small form unmanned aerial vehicles. 10910, 1091014 (2019)
38. Long, G.L., Liu, X.S.: Theoretically efficient high-capacity quantum-key-distribution scheme. Phys. Rev. A. 65(2), 32302 (2002). https://doi.org/10.1103/PhysRevA.65.032302
39. Qi, R., et al.: Implementation and security analysis of practical quantum secure direct communication. Light Sci. Appl. 8(22), 12 (2019). https://doi.org/10.1038/s41377-019-0132-3


**How to cite this article:** Zhang, P., et al.: Timing and synchronisation for high-loss free-space quantum communication with Hybrid de Bruijn Codes. IET Quant. Comm. 1–10 (2021). https://doi.org/10.1049/qtc2.12019