QKD on a Board Limited by Detector Rates in a Free-Space Environment

Alan Mink and Joshua C Bienfang
National Institute of Standards and Technology (NIST),
Gaithersburg, MD, USA
amink@nist.gov, joshua.bienfang@nist.gov

***Abstract*—We discuss a high-speed quantum key distribution (QKD) system with the protocol infrastructure implemented on a printed circuit board that can operate with various photonic subsystems. We achieve sub-nanosecond resolution with serial data receivers operating up to 2.5 Gb/s. Data processing bottlenecks are avoided with pipelined algorithms and controlled data flow implemented in a field-programmable gate array. This eliminates processing on the attached computer and frees CPU cycles for related activities, such as key management and system monitoring. Operating in a laboratory setting, we tested the QKD boards up to their maximum 2.5 GHz transmission rate, and found that under low-link-loss, high-count-rate conditions, timing jitter in the single-photon detectors imposed critical limitations to the maximum achievable throughput.**

Key words: quantum communication, QKD, programmable instrumentation, gigahertz signals

I. INTRODUCTION

The current generation of quantum key distribution (QKD) [1] systems has achieved Mb/s of privacy amplified (PA) key [5,17]. This has been accomplished with Gb/s quantum channels sustained by hardware for data handling and time binning of quantum-channel signals and associated sifting operations. As researchers pursue the next generation of QKD systems that can sustain Gb/s of PA key [4], highly optimized and parallel implementations will be required to handle QKD post-processing as well. While it may not be productive to further increase the quantum channel transmission rate, progress is being made in multiplexing and other photonic configurations [12,18] to achieve Gb/s sifting rates.

Our QKD research focus has been speed. Our initial testbed [2] was designed around a free-space system using a 1.25 GHz transmission rate, which resolves to 800 ps time bins. Although we were able to attempt to send a single photon in each of those time bins, setting our attenuated-laser sources at a mean photon number of 0.1 yields, on average, actual photons in one of every 10 time bins, an average photon emission rate of 125 MHz. Shortly following this free-space testbed, we developed a similar fiber based testbed [16] operating at the same speeds. Anticipating multiple photonic subsystems and quantum channels, we originally planned a common infrastructure for timing, framing, sifting and post-processing (reconciliation [13] and privacy amplification). To handle 800 ps time bins, we designed hardware to manage timing, framing and sifting. This reduces the GHz data rates to MHz data rates for post-processing, which we originally thought could be handled in a sustained fashion by software on a computer. We found that this approach worked for data rates up to about 1 Mb/s for PA data. Because our hardware had a capacity of greater than 30 Mb/s of sifted key and our QKD systems were producing up to 4 Mb/s of sifted keys [17], we developed an enhanced version of our hardware that would also implement post-processing, and thereby increase the PA key rate. The result was a hardware implementation able to operate at 2.5 GHz, using 400 ps time bins, with an output capacity of up to 12 Mb/s of PA key. We note that although it is feasible to distribute post-processing over multiple software instances, this approach was deemed to be too cumbersome for practical deployment and the compact hardware approach was more appealing.

Here we report [20] on experiments with this 2nd generation hardware infrastructure. We attempted to determine some of our QKD system limitations in a laboratory testing environment. In so doing, we discovered that in our implementation, faster transmission rates did not result in significantly faster PA key rates, primarily due to jitter in our single-photon detectors. The remainder of this paper will outline our hardware designs followed by our experimental free-space QKD configuration and the performance we observed.

II. HARDWARE INFRASTRUCTURE

Our 1st generation hardware to manage the timing, framing and sifting was a pair of custom designed printed circuit boards (PCBs), see Fig. 1, that included a field programmable gate array (FPGA) for processing, GHz serializer/deserializer (SERDES) chips for communication and a PCI interface to exchange data with the computer. FPGAs have about an order of magnitude slower clock rate than CPUs, but allow a designer to define arbitrary complex logical operations with an extensive level of parallelism that can make up for the lower clock rate. Furthermore, FPGAs are not hobbled by random operating system interrupts and other background processing tasks that make guaranteeing a fixed number of compute cycles in a given time interval impossible. SERDES are the foundation of high-speed transceivers. They convert between a parallel data stream at a lower data rate and a serial data stream at a higher data rate. For example, between a 10-bit parallel data stream at 125 MHz and a serial data stream at 1.25 GHz. The higher speed serial data stream is for transmission, while the lower speed parallel data stream is for processing on the FPGA. SERDES also provide an important clock-recovery function

ISBN: 978-1-61208-303-2

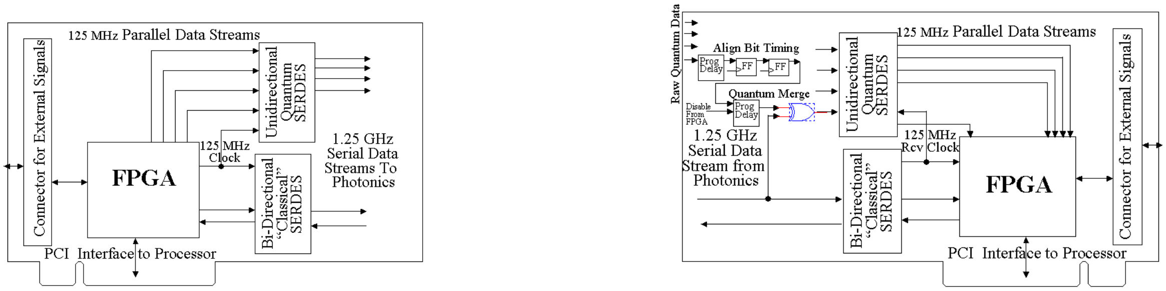
125 MHz Parallel Data Streams
Connector for External Signals
Unidirectional Quantum SERDES
FPGA
125 MHz Clock
1.25 GHz Serial Data Streams To Photonics
Bi-Directional "Classical" SERDES
PCI Interface to Processor
Raw Quantum Data
Align Bit Timing
Prog Delay
FF
FF
Quantum Merge
Disable From FPGA
Prog Delay
1.25 GHz Serial Data Stream from Photonics
125 MHz Rcv Clock
125 MHz Parallel Data Streams
Unidirectional Quantum SERDES
Bi-Directional "Classical" SERDES
FPGA
Connector for External Signals
PCI Interface to Processor

Figure 1. 1st generation PCB Functional Block Diagrams of Alice (left) and Bob (right), four quantum channels and one classical channel.

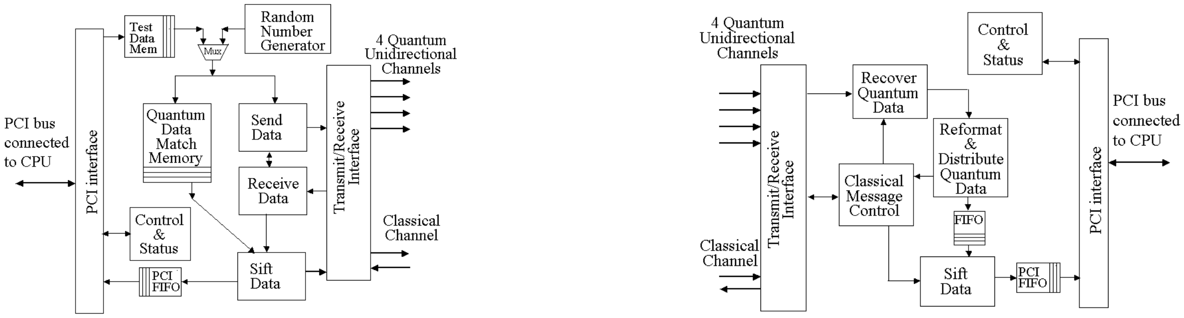

Figure 2. 1st generation PCB Logical Modules of Alice (left) and Bob (right).

that allows the receiver to synchronize to the clock of the transmitter.

To explain the operation of the FPGA firmware, we walk through the flow of the logical modules shown in Fig. 2. The Random Number Generator module on Alice's FPGA generates two bit-streams of pseudo random data, each at up to 1.25 Gbit/s; one stream for the bit value and the other for the basis. Their 2-bit combinations define the four polarization states transmitted on the quantum channel. These streams are temporarily stored in the Match Memory as well as passed to the Send Data modules where each 2048 bit pairs are grouped into a packet. Each packet is then passed to the Transmit/Receive module where they are synchronously used to control the photonics that send signals to Bob on the quantum channel and a "Sync" message on the classical channel. These electrical signals are sent from the PCB to the photonics, where they are shaped and converted to optical signals for the classical and quantum channels.

When a "Sync" message is received by the Transmit/Receive module in Bob's FPGA, it begins the capture of one packet's worth of data from the Quantum channel detectors. At this point the photonics have separated the photon arrival stream into four separate electrical signals, corresponding to the four possible measurement outcomes. Although the first transmission event leaves Alice at the same time as the first bit of the "Sync" message, it can arrive sometime later than the "Sync" message because quantum-channel signals follow a different path than the classical-channel signals. We measure this channel delay and specify its value, via the PCI interface, to the FPGA to provide the necessary compensation. For each detector, a packet's worth of time-bin samples are captured and are passed to the Recover Quantum Data module where they are aligned and then searched for rising edges that denote a detection event. The location within the quantum packet and the associated detector (i.e., tagged time bin, basis and value) are passed to the Reformat & Distribute Quantum Data module. This module reformats the data into a set of triples consisting of time bin, basis and bit value. For each packet, this set is temporarily stored in a FIFO and also passed to the Classical Message Control module where time bin and basis information, a detection pair, is sent back to Alice for sifting.

When Alice's Receive Data module gets a packet's detection pairs, which could be empty, it passes that information to the Sift module. The Sift module compares the basis value of each pair against the stored value in the Quantum Data Match Memory. If they match, then the bit value stored in the Match Memory is placed in the PCI FIFO forming Alice's stream of ordered Sifted bits. The associated state stream is then deleted from Alice's temporary database and a copy of the matching detection pair is also sent back to Bob as an acknowledgement. When Bob's Classical Message Control module receives the acknowledge list, Bob passes that list to its Sift module, which compares it against the list in its Temp FIFO and discards all entries that are not on the acknowledge list (i.e., those with incorrect basis). For those items that are on the list, the bit value is placed in the PCI FIFO forming Bob's stream of ordered Sifted bits. These Sifted bits are passed to an application program running on the CPU via a device driver in the operating system through a DMA (Direct Memory Access) transfer. DMA is a fast memory transfer that does not require CPU intervention, thus allowing the CPU to continue computation during the transfer.

As mentioned above, this initial design was hampered by the software post-processing speed, limited on-chip memory

ISBN: 978-1-61208-303-2

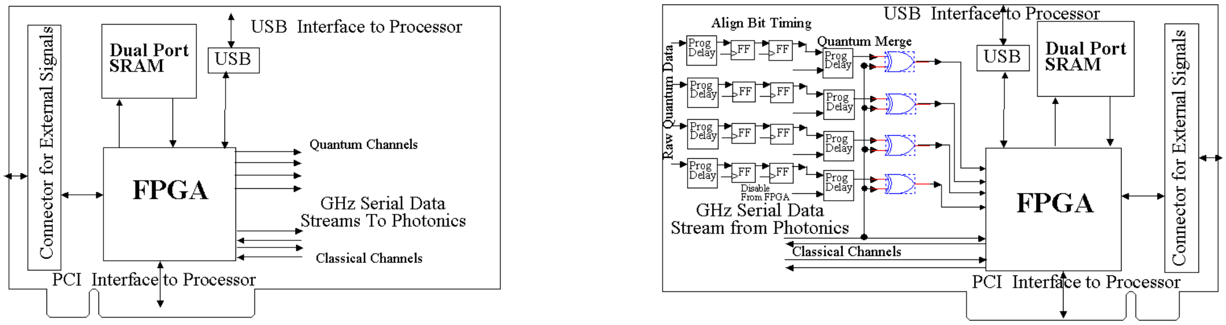
Connector for External Signals
Dual Port SRAM
USB
USB Interface to Processor
FPGA
Quantum Channels
GHz Serial Data Streams To Photonics
Classical Channels
PCI Interface to Processor
Align Bit Timing
Raw Quantum Data
Prog Delay
FF
Quantum Merge
Disable From FPGA
GHz Serial Data Stream from Photonics
Classical Channels
USB Interface to Processor
USB
Dual Port SRAM
FPGA
Connector for External Signals
PCI Interface to Processor

Figure 3. 2nd Generation PCB Functional Block Diagrams of Alice (left) and Bob (right), four quantum channels and two classical channels.

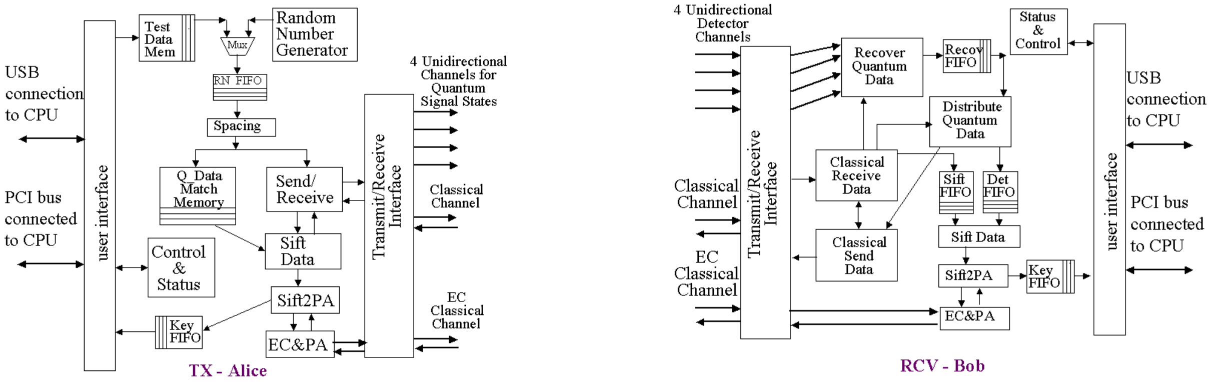

Figure 4. 2nd Generation PCB Logical Modules of Alice (left) and Bob (right).

that degraded performance as the distance between Alice and Bob increased, and the inability to increase the transmission rate that controls the time-bin temporal resolution. Our 2nd generation hardware, see Fig. 3, incorporated all the functionality of our 1st generation as well as functionality to overcome these limitations [10]. To include the post-processing algorithms we upgraded to a newer FPGA, ten times larger. This upgrade also provided a set of on-chip SERDESs, whose transmission speeds were faster and programmable, thus offering a higher transmission rate and better time-bin resolution. By eliminating the separate SERDES chips, we had space on the PCB for an additional memory chip to buffer data for longer round trip times between Alice and Bob.

Although the post-processing logical block, “EC&PA”, in Fig. 4 seems like a small addition to Fig. 2, it represents the vast majority of the new FPGA implementation. For error correction, we are using a variant of the Cascade reconciliation algorithm [13]. We have also implemented the low density parity check (LDPC) error correction algorithm [14,11]. Cascade is an interactive algorithm that requires multiple round trips to refine information on the data to be corrected. LDPC is a one-way algorithm that requires a single transmission of correction data, but different error correction structures are required for different error rates and changing these structures in hardware is slow and inconvenient. Cascade requires about 1 to 2 bytes of memory per bit of data to be corrected, whereas LDPC requires about 20 to 30 bytes of memory – an order of magnitude more. Our FPGA implementation of Cascade was about twice as fast as our LDPC FPGA implementation. Even accounting for the latency associated with a link length of 200 km between Bob and Alice, Cascade was still faster. Because of the FPGA memory limitations, we were able to install four parallel tasks (threads) of Cascade, but only two threads of LDPC. Thus we chose Cascade because of the speed advantage.

The sifting process presents the reconciliation algorithms with a set of ordered bits that do not need any identification with either Alice or Bob. The peer reconciliation algorithms are asymmetrical, but are independent of Alice and Bob, and we designate them as Active and Passive. As in our software implementation, we use multiple parallel threads. The memory requirement of Active is more than that of Passive. To conserve the limited memory of the FPGA, we allocate an equal number of both Passive and Active threads on Alice and the opposite combination of threads on Bob.

Each Active and Passive pair implements reconciliation in three phases. Each phase requires at least one round trip communication to exchange information. Phase 1 and 3 are executed once, while phase 2 may be repeated. A summary follows:

1. Active and Passive identically randomize their bits, divide them into many disjoint groups and compute parity for each group. A group is less than 100 bits and

greater than 5 bits. Passive sends its set of parity to Active, who uses it to estimate the error rate and to identify groups that need correction. If the estimate is too high, Active and Passive discard the data, wait for new data and restart **phase 1**. If the estimate is low enough to process, Active computes a Hamming code on each group to be corrected and sends that set of Hamming codes to Passive. Passive decodes each Hamming code and affects correction where possible. Then go to **phase 2**.

2. Active and Passive identically randomize their bits, divide them into many disjoint groups and compute parity for each group. As the remaining error rate decreases, the size of a group increases. Passive sends its set of parity to Active, who uses it to estimate the remaining error rate and to identify groups that need correction. If the estimate is below a threshold, Active and Passive go to **phase 3**. Otherwise if the probability is above the threshold, Active computes a Hamming code on each group to be corrected and sends that set of Hamming codes to Passive. Passive decodes each Hamming code and affects correction where possible. Repeat **phase 2** until the remaining error probability drops below the threshold. If the maximum repetitions have been exceeded, discard the data and, wait for new data and restart at **phase 1**.
3. Execute a special final correction pass. Active computes a Hamming code on each group to be corrected and sends that set of Hamming codes to Passive. Passive decodes each Hamming code and affects correction where possible. Where not possible, discard that group and send a list of discarded groups to Active. Active and Passive send their bits to the next stage for verification and PA. Then they wait for new data and restart at **phase 1**.

This implementation of Cascade repeatedly executes a few operations that can reuse the same memory and the same logic resources. This results in a compact and efficient FPGA implementation. Since keeping the Active and Passive set of bits aligned is essential, randomizing those bits must be done exactly the same by both. This is accomplished by using a pseudo-random generator with Active and Passive using the same seed. Since the randomization is used to mix up the bits and expose errors, there is no need to keep that seed secret, although one can. The information exposed during Cascade is the sum of the sets of parity bits plus the sum of the sets of Hamming codes. Thus we keep track of this total, 1 bit for each parity bit and $\log_2(n)$ for each n-bit Hamming code. This total represents the reduction due to error correction during PA.

The bulk of the parallelism comes from multiple Cascade threads. Because of the interactive nature of Cascade, there is not much parallelism within a thread, although the communication latency can be mitigated by overlapping the waiting time with computations.

The now corrected data is accumulated for PA, but first a hash code signature of that data is computed and exchanged for comparison. These bits contribute an additional reduction during PA. If the hash signatures differ, the data is discarded. If the signatures are the same, the PA algorithm is invoked and the resulting PA bits are passed to the CPU.

The resulting capacity performance of this 2nd generation infrastructure is about 12 Mb/s for a QBER of 1 % and about 10 Mb/s for a QBER of 2 %. This is an order of magnitude faster than our 1st generation hardware. These rates were obtained using simulated QKD data, since our QKD photonic systems could not produce high enough data rates to stress this infrastructure, as we discuss below.

III. EXPERIMENTAL TESTS

To test the performance of the 2nd generation QKD hardware infrastructure in conjunction with an actual quantum-channel physical layer, we use the BB84 system described in [15]. To minimize link loss and allow the fullest range of throughputs for testing the hardware, the system is setup in a laboratory setting with a 1 m free-space path between Alice and Bob. The classical channel for timing and sifting operates at 2.5 Gb/s and the classical channel for post-processing operates at 1.25 Gb/s, both are at 1550 nm over 15 m of optical fiber, and the quantum channel operates at transmission rates up to 2.5 GHz at 851.4 nm with attenuated gain-switched vertical-cavity surface-emitting lasers (VCSELs) producing < 50 ps optical pulses. The narrow-band interference filters used to block background solar photons described in [15] were removed from Bob's receive aperture. Opaque enclosures and high-transmissivity 10 nm bandpass filters at the single-photon detectors were used to suppress background light in the lab below the dark count level of the detectors. As described in [15], the four silicon single-photon avalanche diode detectors were modified for improved timing resolution, and exhibit a full-width at half-maximum of roughly 200 ps for count rates up to about 1 MHz. Link losses up to -27 dB are simulated by inserting neutral density filters in the 1 m path between Alice and Bob. The protocol implemented in the hardware infrastructure is not yet configured for decoy-state QKD [9], meaning that the high link losses investigated in this experimental test exceed the operational range of the current configuration. Nonetheless, it was deemed valuable to test the hardware's data-processing capabilities over a wide range of throughputs, particularly those at the lowest link losses and highest count rates.

The quantum channel SERDES operate at 2.5 GHz, providing 400 ps detection time bins. To operate at lower transmission rates, we simply spaced transmission events by 1, 2, or 4 clock cycles. At the lower transmission rates, the hardware's timing resolution allowed us to operate in either a gated mode, in which only detection events that occur in the transmission time bin are retained for sifting, or an ungated mode, in which events that occur at any time during a transmission period of 2 or 4 time bins are retained. For example, at 625 MHz there would be one 400 ps transmission time bin as well as three additional 400 ps time bins before the next possible transmission time bin. Gated mode retains events in only the transmission time bin while ungated mode retains events occurring in any of the four time bins.

ISBN: 978-1-61208-303-2

The results of the experimental trials are shown in Figs. 5-7. In all cases, higher transmission rates resulted in higher sifted-bit rates, as expected. However, as shown in Fig. 6, lower link attenuations caused the QBER to rise significantly, particularly for a transmission rate of 2.5 GHz. The resulting error corrected and PA throughput is shown in Fig. 7, where it is clear that, for this system, transmission rates of 1.25 GHz or 625 MHz outperform 2.5 GHz at most attenuations. This is because the QBER detriment induced by detector jitter at high average count rates outweighs any benefit in additional sifted bit rates that may be gained by operating at 2.5 GHz transmission rates.

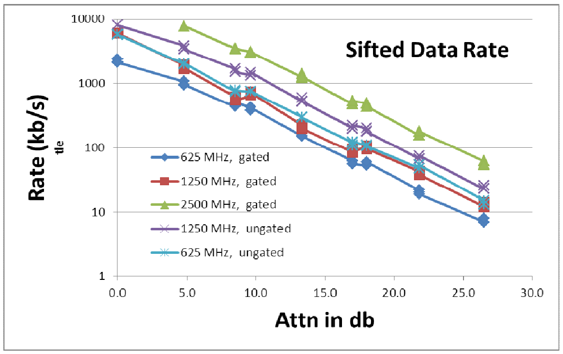

Figure 5. Sifted Data Rate Measurements.

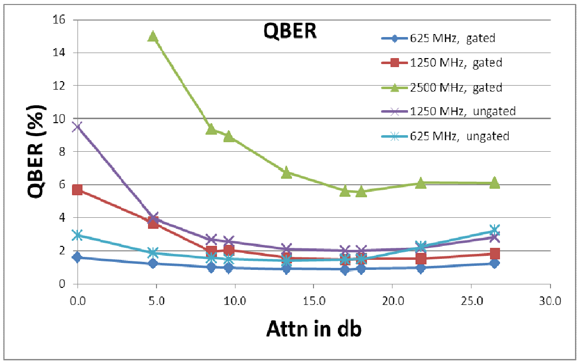

Figure 6. QBER Measurements.

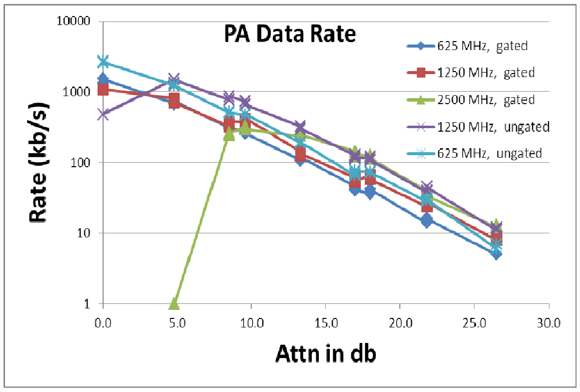

Figure 7. Privacy Amplified Data Rate Measurements.

The rise in QBER at attenuations below 10 dB reflects the increase in detector jitter with count rate, even with the detector modifications described in [15]. At these low link losses and high transmission rates, each detector is counting at rates well into the MHz range, and the timing resolution of the detectors is significantly degraded. At these high count rates and sifted bit rates, the detector jitter become so great that there is a significant probability that a detection event will occur in a clock cycle later than the one in which it was transmitted, driving up the QBER. Above a QBER of 11 %, the post-processing algorithm cannot distill bits from the sifted string.

The highest PA throughput was achieved at the more moderate transmission rate of 625 MHz. With about 5.7 Mb/s of sifted key and a QBER close to 3 %, the PA rate was about 2.6 Mb/s. The highest sifting rate of about 8 Mb/s was achieved at a transmission rate of 1.25 GHz, but because of a QBER of about 9.5 %, the PA rate was about 480 Kb/s. Unfortunately these bit rates are well within the performance range of the hardware, and the physical layer was not able to produce data at rates sufficient to stress the hardware beyond its capabilities. With a detector dead time, τ, of 50 ns, our secure sifting limit [3] is about 10 Mbits/s = 1/(2*τ). Even within our laboratory environment we were not able to achieve that rate.

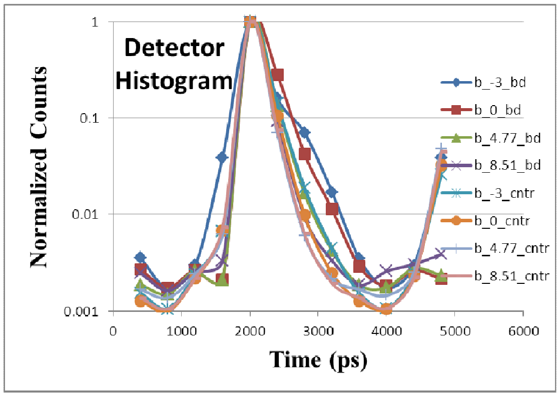

Figure 8. Samples of single photon detector histogram.

Our PCB can also capture detection histogram data with a 400 ps time-bin resolution and we compared that to a commercial time-correlate single-photon counting system (TCSPC) with 4 ps time-bin resolution. While transmitting at 312.5 MHz, we measured each detector separately to determine the jitter effects in the time bins other than the one in which we transmitted. For 312.5 MHz, we are transmitting in every 8th 400 ps time-bin. Theoretically we should only see detection events in every 8th time-bin, but it is well known that these detection histograms have a long tail caused by jitter, as can be seen in the example shown in Fig. 8. Although the TCSPC yielded cleaner measurements with deeper lows, our board measurements were reasonably close, but neither was able to predict the QBERs we measured.

IV. GB/S QKD

As researchers pursue the next QKD level of Gb/s PA key [4], highly optimized and parallel implementations will be required to handle QKD post-processing. The current replacement for Cascade seems to be LDPC that can asymptotically approach the Shannon limit. LDPC and

PA are one-way post-processing algorithms that are coarse grained computations in the sense that a given data set does not need to communicate until a solution is obtained. Thus, each data set can be assigned to a separate independent computation engine and each result could be collected sequentially at completion to maintain their order. Maintaining synchronization between Alice and Bob's bits is required throughout these operations. Up to 100 Mb/s of LDPC error correction performance [6] has been reported for a graphics processing unit (GPU) implementation. Thus 5 computers with 2 GPUs each could error correct at a Gb/s rate. LDPC performance of 47 Gb/s [19] has been reported for a custom chip implementation. It is not clear whether this chip could operate with structures appropriate for QKD, but that chip indicates that such designs are possible.

Similar implementation characteristics apply to PA along with a rather large data set requirement for efficient PA key generation ratios [8]. Brute force algorithms are of $O(n^2)$ complexity, while efficient algorithms based on FFTs are of O(n*log(n)) complexity. FFT performance of about 100 GFlops [7] for 2^{20} elements has been reported for a GPU implementation, which we estimate to be about 1/8 to 1/4 Gb/s privacy amplification rate. Thus 2-4 computers with 2 GPUs each could privacy amplify at a Gb/s rate.

V. CONCLUSION

We have discussed two generations of hardware infrastructure supporting QKD photonic operations. Our newest infrastructure is capable of producing 12 Mbits/s of PA key at 1 % QBER, but because of limitations in our detectors and increased jitter at high data rates we were not able to approach that limit. Furthermore, moving to higher photon transmission rates did not provide for higher PA key rates, because of higher QBERs due to jitter. Attaining Gb/s QKD rates will require specially crafted hardware with high levels of parallelism.

REFERENCES

[1] C. H. Bennet and G. Brassard, "Quantum Cryptography: Public key distribution and coin tossing", Proc of the IEEE Intern'l Conf on Computers, Systems, and Signal Processing, Bangalore, India, Dec. 1984, pp 175-179.

[2] J.C. Bienfang, et al. "Quantum key distribution with 1.25 Gbps clock synchronization", Optics Express. Vol. 12 (9), May 3, 2004, pp 2011-2016.

[3] V. Burenkov, B. Qi, B. Fortescue, and H.-K. Lo, "Security of high speed quantum key distribution with finite detector dead time", arXiv.org:arXiv:1005.0272, 3 May 2010.

[4] DARPA-BAA-12-42, "Quiness: Macroscopic Quantum Communications", May 15, 2012. <https://www.fbo.gov/index?s=opportunity&mode=form&id=6a3a61d577305f71d9be268925c4b201&tab=core&_cview=0> (accessed 5/29/2013)

[5] A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, and A. J. Shields, "Continuous operation of high bit rate quantum key distribution", Appl. Phys. Ltr. **96**, 161102, Apr 19, 2010; <http://dx.doi.org/10.1063/1.3385293> (accessed 5/29/2013)

[6] G. Falcao, V. Silva, and L. Sousa, "How GPUs can outperform ASICs for fast LDPC decoding". Proc. of the 23rd Intern'l Conf on Supercomputing, ACM, 2009, pp 390–399.

[7] N. K. Govindaraju, B. Lloyd, Y. Dotsenko, B. Smith, and J. Manferdelli, "High Performance Discrete Fourier Transforms on Graphics Processors", Proc. ACM/IEEE Conf on Supercomputing, Austin, TX, Nov. 2008, pp 1-12.

[8] M.Hayashi and T.Tsurumaru, "Concise and tight security analysis of the Bennett-Brassard 1984 protocol with finite key length", *New Journal of Physics* **14,** 093014, Sept. 2012. <http://iopscience.iop.org/1367-2630/14/9/093014/> (accessed 5/29/2013)

[9] H.K. Lo, X.F. Ma, and K. Chen, "Decoy state quantum key distribution", Phys. Rev. Lett. 94, 230504 , June 2005.

[10] A. Mink, "Custom hardware to eliminate bottlenecks in QKD throughput performance", Proc. SPIE: Optics East 07, 6780, 678014-1, Boston, MA, Sept. 2007.

[11] A. Mink and A. Nakassis,"LDPC for QKD Reconciliation", The Computing Science and Technology International Journal, Vol. 2, No. 2, June, 2012, ISSN (Print) 2162-0660, ISSN (Online) 2162-0687, June, 2012, <http://www.researchpub.org/journal/cstij/number/vol2-no2/vol2-no2-1.pdf> (accessed 5/29/2013)

[12] J. Mora, W. Amaya, A. Ruiz-Alba, A. Martinez, D. Calvo, V. Muñoz, and J. Capmany, "Simultaneous transmission of 20x2 WDM/SCM-QKD and 4 bidirectional classical channels over a PON", Opt. Express 20, 16358-16365, July 2012.

[13] A. Nakassis, J. Bienfang, and C. Williams, "Expeditious reconciliation for practical quantum key distribution", Proc. SPIE: Quantum Information and Computation II, Proc. SPIE 5436, Aug. 2004, pp 28-35.

[14] A. Nakassis and A. Mink, "LDPC error correction in the context of Quantum Key Distribution", Proc. SPIE: Defense Security & Sensing, Balt., MD, Apr. 2012.

[15] A. Restelli, J.C. Bienfang, C.W. Clark, I. Rech, I. Labanca, M. Ghioni, S. Cova, "Improved Timing Resolution Single-Photon Detectors in Daytime Free-Space Quantum Key Distribution With 1.25 GHz Transmission Rate", IEEE J. Sel. Top. Quantum Electron. 16, Sept. 2010, pp 1084–1090.

[16] X. Tang, L. Ma, A. Mink, A. Nakassis, B. Hershman, J. Bienfang, R. F. Boisvert, C. Clark, and C. Williams, "High Speed Fiber-Based Quantum Key Distribution using Polarization Encoding", Proc. of SPIE Optics and Photonics Conf, San Diego, CA, Vol. 5893, July 2005.

[17] X. Tang, L. Ma, A. Mink, T. Nakassis, H. Xu, B. Hershman, J. Bienfang, D. Su, R. Boisvert, C. Clark, and C. Williams, "Quantum Key Distibution System Operating at Sifted-key Rate Over 4 Mbits/s," SPIE Defense & Security Symp, Orlando, FL, Vol. 6244-25, Apr 2006, pp. 62440P-1 -7.

[18] N. Walenta, A. Burg, J. Constantin, N. Gisin, O. Guinnard, R. Houlmann, C. Ci Wen Lim, T. Lunghi, and H. Zbinden, "1 Mbps coherent one-way QKD with dense wavelength division multiplexing and hardware key distillation", QCRYPTO2012, Singapore, Sept 2012.

[19] Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic. "A 47 Gb/s LDPC Decoder with Improved Low Error Rate Performance". Symp on VLSI Circuits, June, 2009, pp 22-23.

[20] The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology.

ISBN: 978-1-61208-303-2