Key module for a novel all-optical network coding scheme

Zhijian Qu (曲志坚), Yuefeng Ji (纪越峰)*, Lin Bai (柏 琳), Yongmei Sun (孙咏梅), and Jia Fu (付 佳)

Key Laboratory of Information Photonics and Optical Communications, Ministry of Education,
Beijing University of Posts and Telecommunications, Beijing 100876, China
**E-mail: jyf@bupt.edu.cn*

Received March 23, 2010

We introduce a novel network-coding scheme that can be implemented in all-optical multicast networks. A simple and successful module based on the all-optical XOR gate is designed to realize the network coding scheme. The module is a key hardware component in realizing the proposed scheme. The working principle and the experimental results of the module are also presented. Experimental results show that the function of the module is sufficient in satisfying the requirements of the proposed network coding scheme.

OCIS codes: 060.4510, 200.4560, 200.3760.
doi: 10.3788/COL20100808.0753.

Network coding has attracted intense scholarly attention since Ahlswede *et al.* first proposed it in 2000[1]. In addition, it has numerous advantages over traditional store-and-forward based routing solutions. For instance, network coding can increase network throughput, reduce network congestion, improve network reliability and security, and provide protection for multicast networks. Generally, linear computations are used in realizing the classical network coding schemes[1,2]. However, without optical-electronic-optical (OEO) conversion, the scheme cannot be realized in the photonic domain, which lacks optical random access memory (ORAM). Thus, the classical network coding scheme cannot be realized without OEO conversion in all-optical multicast networks. To introduce beneficial network coding into all-optical multicast networks, the network coding scheme that can be realized in the photonic domain is worth investigating. To date, only a few published studies have demonstrated the advantages of introducing network coding into optical networks.

Kamal used network coding over p-cycles to accomplish $1+N$ protection in optical mesh networks[3]. Kim *et al.* proposed a network coding-based two-stage method to achieve a minimum-cost and fault-tolerant multicast in optical multicast networks[4]. However, the effective implementation of their network coding schemes requires OEO conversion. If the signals undergo OEO conversion at each coding node, they can be buffered and processed for the encoding operations. Thus, the traditional network coding scheme can be realized in optical networks. However, this opaque network coding approach inhibits many benefits of optical multicast. In addition, more time is required to perform encoding operations in the coding nodes. Consequently, the coding nodes become the bottleneck of the multicast networks. More importantly, high speed optical signals may be lost in the coding nodes because of the lack of ORAM and the low speed encoding operation in the electronic domain. In this case, the optical multicast performance is bound to be impaired.

Menendez *et al.* used bitwise XOR network coding to protect optical multicast networks[5]. Belzner *et al.* used XOR network coding to reduce packet loss ratio as well as queuing delay to address any congestion in passive optical networks[6]. The XOR network coding can easily be realized in the photonic domain. However, from the perspective of network coding, such bitwise XOR coding is too restrictive for the binary finite field. In other words, the XOR network coding lacks universality.

Manley *et al.* investigated the realization of network coding without OEO conversion in the optical layer. They proposed an architecture supporting the encoding operations based on the Galois fields GF(2^m) normalization step[7]. The architecture contains two serial scalar multiplication and addition units and a combined GF(2^m) normalization unit. Serial-to-parallel and parallel-to-serial converters are required in the architecture. Moreover, in the serial scalar multiplication and addition unit, 2 m SOAs and in most m all-optical XOR gates are required to accomplish the function. Thus, their architecture is deemed too complex and difficult to realize.

With the development of all-optical logic devices[8−11], such as all-optical XOR gates[10] and all-optical shift registers (OSR)[11], it is now feasible to realize network coding using logical computations in the photonic domain. In this letter, we focus on realizing a two-channel network coding scheme without OEO conversion in all-optical multicast networks[12]. A novel network coding scheme based on logical computations is thus presented in order to achieve this goal. A key module is designed and experimentally demonstrated to realize the proposed network coding scheme. The proposed scheme has a higher degree of universality than the XOR network coding scheme. It is also simpler than the method proposed by Manley *et al.*

We propose a novel two-channel network coding scheme that can be realized in the photonic domain. In this scheme, we use bit left shifting and logic XOR operations, instead of the multiplication and addition operations that are employed in the traditional network coding scheme. To support the two network coding operations, bit left shifting and logical XOR operations, we designed the coding vector sets: $U = \{\mathbf{p}_n = (1\ 2^n)^{\mathrm{T}} |\ n{\in}N\ \}$ and $V = \{\mathbf{q}_m{=}(2^m\ 1)^{\mathrm{T}} |\ m{\in}N,\ m{>}0\}$, where $\mathbf{p}_n$ and $\mathbf{q}_m$ served as the column vectors and N represented the set of natural numbers. The coding vectors of the network coding scheme were chosen from the column vector set U or V.

U and V are linearly independent sets, while $\mathbf{p}_n$ and $\mathbf{q}_m$ are linearly independent vectors. Thus, $m+n+1$ coding vectors can be chosen for encoding. In practice, m and n can be determined by the number of coding nodes on the multicast tree. For example, if there are t coding nodes on the multicast tree, m and n can be determined using the formula $m+n+1=t$, and $\min(|m-n|)$ is best. The purpose of designing such coding vectors is to ensure that the multiplication operations of the network coding scheme can be implemented through bit left shifting because multiplication by 2^n can be achieved by left shifting the multiplicand n bits. To prevent an overflow of the multiplicand by bit left shifting, the highest n bits of the multiplicand are reserved and initialized as "0".

The coding process of the proposed scheme is different from traditional processes. In traditional methods, coding vectors are chosen from a linearly independent vector space. As such, received data are encoded directly by the linearly independent coding vectors in the coding node, and all the sinks can recover the original data by linear decoding operations. However, this method is no longer valid in the novel coding scheme. Both U and V are vector sets rather than vector spaces; hence, when the two original signals are encoded more than once, the vector of the coding result will no longer be an element of U or V. Consequently, the sinks cannot decode.

For example, in Figs. 1(a) and (b), binary signals "a" and "b" are the original signals emitted by the source. The symbol "$*$" represents the multiplication operation. The binary signals "$2*a\oplus b$" and "$a\oplus 2*b$" are the input signals of the coding node. They are the encoded results of the original signals by the coding vectors $(2\ 1)^{\mathrm{T}}$ and $(1\ 2)^{\mathrm{T}}$ in the respective upstream coding nodes. The encoding operations can be expressed as matrix multiplication $(a\ b)*(2\ 1)^{\mathrm{T}}=(2*a\oplus b)$ and $(a\ b)*(1\ 2)^{\mathrm{T}}=(a\oplus 2*b)$, respectively. Here, the traditional addition operation is replaced by the XOR operation. From the perspective of hardware realization, the signal "$2*a\oplus b$" can be obtained by left shifting "a" one bit and then XOR operating it with "b"; the signal "$a\oplus 2*b$" can be achieved by the same method.

In Fig. 1(a), the two input signals are again encoded by the coding vector $(1\ 4)^{\mathrm{T}}$ in the coding node, and the encoding result "$2*a\oplus b\oplus 4*a\oplus 8*b$" is the output.

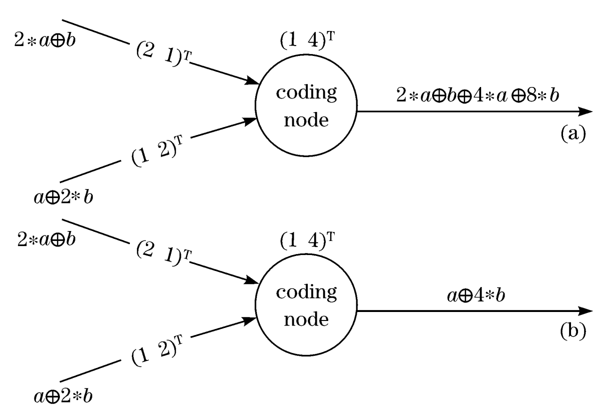

Fig. 1. (a) Traditional and (b) novel coding processes.

The expression of the encoding operation is $((2*a\oplus b)\ (a\oplus 2*b))*(1\ 4)^{\mathrm{T}}=((2*a\oplus b)\oplus(4*(a\oplus 2*b)))=(2*a\oplus b\oplus 4*a\oplus 8*b)$. Here, the coefficient vector of the coding result is four-dimensional because the expression $2*a\oplus b\oplus 4*a\oplus 8*b\neq 6*a\oplus 8*b$ is true in most cases of logical XOR operation. In fact, the dimension of the coefficient vector of a coding result can range from 1 to n ($n>=2$), and the value of n depends on the frequency of coding. The foregoing analysis shows that the sinks cannot decode the signals when the two original ones are encoded more than once.

To ensure that the original signals are recovered from the two coding results in the sinks, the coefficient vector of each coding result must be kept two-dimensional. To ensure this, the encoded signal should first be decoded in each coding node. In this way, two original signals can be recovered in each coding node. Subsequently, the recovered original signals are recoded by the coding vector assigned to the coding node in each coding node. These operations can guarantee that the coefficient vector of each coding result is two-dimensional and that the sinks can decode the result.

For example, the coding node presented in Fig. 1(b) is equipped with decoding ability. The input binary signals "$2*a\oplus b$" and "$a\oplus 2*b$" are not directly encoded by the coding vector $(1\ 4)^{\mathrm{T}}$, but are first decoded to recover the original signals "a" and "b". The coding node then encodes "a" and "b" by the coding vector $(1\ 4)^{\mathrm{T}}$ and produces as output the two-dimensional result "$a\oplus 4*b$". Following this, the output result becomes easy to handle with the downstream nodes.

We present the process of decoding the two encoded input signals in a coding node. To maintain the universality of the illustration, we assume that the two input signals are "$2^n*a\oplus b$" and "$a\oplus 2^m*b$" with coding vector $(2^n\ 1)^{\mathrm{T}}$ and $(1\ 2^m)^{\mathrm{T}}$, respectively. They are then encoded by the upstream coding node, where $n>=0$, $m>1$. In order to decode the input signals in the coding node, the signal "$2^n*a\oplus b$" is left shifted m bits and becomes "$2^{m+n}*a\oplus 2^m*b$" (to prevent an overflow of the left shifting operation, the highest $m+n$ bits of the data segment of each packet must be reserved for shifting when created). Thereafter, signal "$2^{m+n}*a\oplus 2^m*b$" and another input signal "$a\oplus 2^m*b$" are XOR operated in an all-optical XOR gate, and the result "$2^{m+n}*a\oplus a$" is obtained. To recover an original signal, the most important step is to recover "a" from the XOR result "$2^{m+n}*a\oplus a$". If "a" is recovered, "2^n*a" can be obtained by left shifting "a" n bits, and "b" can be obtained by XOR operating "2^n*a" and "$2^n*a\oplus b$". When the two original signals are recovered, the coding vectors assigned to the coding node are used to encode the two recovered signals. The network coding operations are completed in this coding node.

Next, we illustrate the principle of recovering the original signal "a" ("a" can represent either of the original signals) from formulas, such as "$2^x*a\oplus a$" ($x>=1$). This principle is vital in realizing the network coding scheme.

The input signal "$2^x*a\oplus a$" is the XOR result of signal "2^x*a" and the original signal "a". Signal "2^x*a" is obtained from "a" by left shifting "a" x bits, thus the lowest x bits of "2^x*a" are "0". In practice, each bit of "$2^x*a\oplus a$" represents the comparison result of two bits of

"a". The two bits interval is x.

To recover signal "a" from "$2^x{*}a\oplus a$", the lowest x bits of "a" must first be obtained. Thereafter, the higher bits are obtained one by one according to the lower bits. The lowest x bits of "a" are equal to the lowest x bits of "$2^x{*}a\oplus a$" because the lowest x bits of "$2^x{*}a$" are "0". The $(x+1)$th bit of "$2^x{*}a\oplus a$" (from low to high) is the comparison result of the first bit and the $(x+1)$th bit of "a". Thus, the $(x+1)$th bit of "a" can be obtained by XOR operating the $(x+1)$th bit of "$2^x{*}a\oplus a$" and the first bit of "a". The other bits of "a" can be obtained by XOR operating the $(x+t)$th bit of "$2^x{*}a\oplus a$" and the tth bit of "a" (t=2, 3,$\cdots$, q, q is the length of the binary signal "a"). The key step of the network coding scheme relies upon this calculation principle. The steps in calculating "a" from "$2^x{*}a\oplus a$" are shown in Fig. 2.

Now, we give an example to illustrate the principle, presuming that "a" is "00010101" (the highest 3 bits are reserved for bit left shifting), thus "$8{*}a$" is "10101000" and "$8{*}a\oplus a$" is "10111101". If we suppose that "a" is unknown, we still know that "a" has been left shifted 3 bits (via the coding vector). Thus, the lowest 3 bits of "$8{*}a$" are known and they are all "0". Therefore, the lowest 3 bits of "a" can be obtained by XOR operating the lowest 3 bits of "$8{*}a\oplus a$" and the lowest 3 bits of "$8{*}a$". The 4th bit of "$8{*}a\oplus a$" represents the comparison result of the 4th bit and the first bit of "a", thus the 4th bit of "a" can be obtained by XOR operating the first bit of "a" and the 4th bit of "$8{*}a\oplus a$". The other bits of "a" can be obtained in the same way. The steps of the calculation process are shown in Fig. 3.

Figure 4 shows the calculation module that can realize the calculation principle. Assume that "$2^x{*}a\oplus a$" enters into the calculation module from port input_1, and the

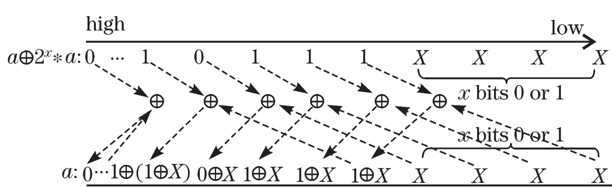

Fig. 2. Calculation principle of the module.

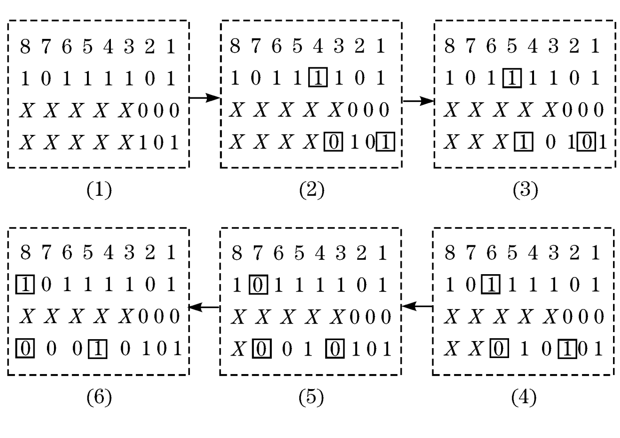

Fig. 3. An example of the calculation principle. From the top to the bottom of each box, the 1st line is the bit number, the 2nd line is "$8{*}a\oplus a$", the 3rd line is "$8{*}a$", the 4th line is "a", and the squares are the XOR results of the circles in each box.

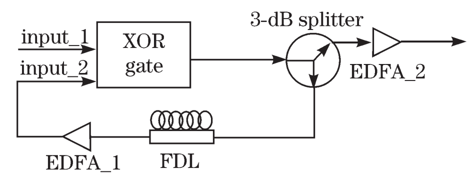

Fig. 4. Structure of the calculation module.

order of the signal passing through the optical XOR gate is from the lowest bit to the highest bit. The output of the XOR gate is duplicated by the 3-dB splitter. One copy is delayed x bits time and goes back to port input_2, while the other copy acts as the output result (signal "a") of the calculation module. After x bits time, the $(x+1)$th bit of "$2^x{*}a\oplus a$" and the first bit of the output delayed by the fiber delay line (FDL) simultaneously enter into the XOR gate. The XOR result of these two bits is the $(x+1)$th bit of "a". The $(x+2)$th bit of "a" is the XOR result of the $(x+2)$th bit of "$2^x{*}a\oplus a$" and the second bit of the output. Other uncalculated higher bits of "a" can be obtained in the same way.

To verify the function of the calculation module, experiment was carried out. The experimental setup was constructed using 3-dB couplers, 1:9 splitters, adjustable fiber delay lines (AFDLs), band-pass filter (BPF), semiconductor optical amplifiers (SOAs), and erbium-doped fiber amplifiers (EDFAs), as shown in Fig. 5. The data rate used was 2.5 Gb/s, and the input optical signals operated at 1550 and 1555 nm, respectively. The XOR gate used in the calculation module was proposed in Ref. [10]. The principle of the XOR gate is based on cross-gain modulation; thus, each input port requires the input signal to be of a single wavelength. To meet the requirement of the XOR gate, the two input signals of the module were carried by different wavelengths (1550 and 1555 nm). Thus, the wavelength of the output was a combination of 1550 and 1555 nm. The field-programmable gate array (FPGA) board was used to convert the XOR result to a single wavelength signal (1555 nm) and to reshape it.

In the experiment, we used the repeated signal "01111011111101000" as the input signal "$a\oplus 8{*}a$". Theoretically, the required output signal "a" is "0000111010101000", and signal "$8{*}a$" is "0111010101000000". The AFDL located outside the XOR gate was used to delay the first bit of the input signal and let it enter the XOR module parallel with the 4th bit of the input signal. Figure 6(a1) shows the input signal; Fig. 6(a2) shows the calculation result, which has not been processed by the FPGA board; and Fig. 6(a3) shows the result signal output from the FPGA board. Figure 6(b) shows the standard eye diagram of the output, which is obtained by directly inputting the theoretical output to the oscilloscope. Figure 6(c) shows the practical eye diagram of the FPGA board output. From the results of the experiment, we can find that the presented calculation module works well. In other words, it can well support the novel network coding scheme in the photonic domain.

In conclusion, we consider the realization of network coding in all-optical multicast networks without OEO conversion. Logical bit shifting and bit XOR operations

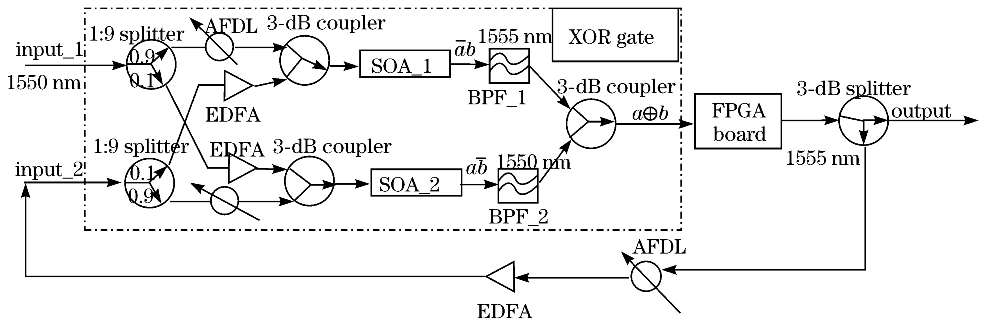

Fig. 5. Experimental setup of the calculation module.

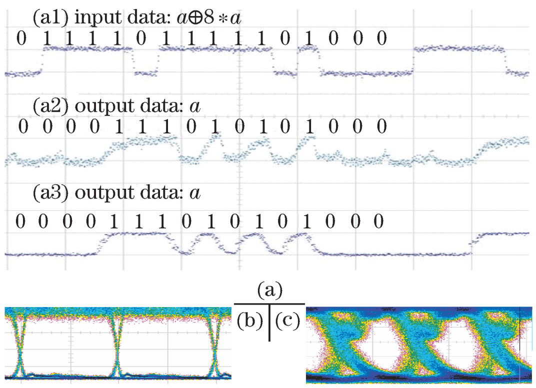

Fig. 6. Experimental results. (a) Waveform of the signals; (b) standard eye diagram of the output; (c) practical eye diagram of the FPGA board output.

are employed to achieve this goal. The proposed network coding scheme has a higher level of universality than XOR network coding because it employs a finite field, which is larger than the binary one. Moreover, a key module of the coding node, which is based on the all-optical XOR gate, is advanced to support the proposed network coding scheme. The function of the module is to recover an original signal from the encoded input signal. This is a key step in realizing network coding in the proposed network coding scheme. Experimental results show that the module works well.

This work was supported in part by the National Natural Science Foundation of China (Nos. 60772024 and 60932004), the National "973" Program of China (No. 2007CB310705), the National "863" Program of China (No. 2009AA01A345), and the Specialized Research Fund for the Doctoral Program of Higher Education, Ministry of Education of China (No. 200800130001).

References

1. R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, IEEE Trans. Inf. Theory **46,** 1204 (2000).
2. S.-Y. R. Li, R. W. Yeung, and N. Cai, IEEE Trans. Inf. Theory **49,** 371 (2003).
3. A. E. Kamal, in *Proceedings of IEEE Globecom 2006 OPN04-3* (2006).
4. M. Kim, M. Médard, and U.-M. O'Reilly, in *Proceedings of OFC 2009* OThO3 (2009).
5. R. C. Menendez and J. W. Gannett, in *Proceedings of OFC 2008* JThA82 (2008).
6. M. Belzner and H. Haunstein, in *Proceedings of ECOC 2009* P6.20 (2009).
7. E. D. Manley, J. S. Deogun, L. Xu, and D. R. Alexander, "Network coding for WDM all-optical multicast" Technical Report TR-UNL-CSE-2009-0007, University of Nebraska-Lincoln (2009).
8. B. Han, J. Yu, L. Zhang, W. Wang, Y. Jiang, A. Zhang, and E. Yang, Chinese J. Lasers (in Chinese) **36,** 2367 (2009).
9. J. Pan and Y. Sun, Chin. Opt. Lett. **6,** 408 (2008).
10. J. H. Kim, Y. M. Jhon, Y. T. Byun, S. Lee, D. H. Woo, and S. H. Kim, IEEE Photon. Technol. Lett. **14,** 1436 (2002).
11. R. McDougall, A. Poustie, G. Maxwell, B. Harmon, J. Reed, P. Townley, M. Harlow, I. Lealman, and L. Rivers, in *Proceedings of ECOC 2005* We2.4.4 (2005).
12. X. Liu, Y. Ji, L. Bai, H. Wang, and Y. Sun, Chin. Opt. Lett. **7,** 983 (2009).