University of Nebraska - Lincoln DigitalCommons@University of Nebraska - Lincoln

CSE Technical reports

Computer Science and Engineering, Department of

4-17-2009

Network Coding for WDM All-Optical Multicast

Eric D. Manley University of Nebraska-Lincoln

Jitender S. Deogun University of Nebraska-Lincoln, jdeogun1@unl.edu

Lisong Xu University of Nebraska-Lincoln, xu@cse.unl.edu

Dennis R. Alexander University of Nebraska-Lincoln, dalexander1@unl.edu

Follow this and additional works at: https://digitalcommons.unl.edu/csetechreports

Part of the Computer Sciences Commons

Manley, Eric D.; Deogun, Jitender S.; Xu, Lisong; and Alexander, Dennis R., "Network Coding for WDM All-Optical Multicast" (2009). *CSE Technical reports*. 10. https://digitalcommons.unl.edu/csetechreports/10

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Technical reports by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Technical Report TR-UNL-CSE-2009-0007 Network Coding for WDM All-Optical Multicast

Eric D. Manley, Jitender S. Deogun, Lisong Xu, and Dennis R. Alexander

Abstract—Network coding has become a useful means for achieving efficient multicast, and the optical community has started to examine its application to optical networks. However, a number of challenges, including limited processing capability and coarse bandwidth granularity, need to be overcome before network coding can be effectively used in optical networks. In this paper, we address some of these problems. We consider the problem of finding efficient routes to use with coding, and we study the effectiveness of using network coding for optical-layer dedicated protection of multicast traffic. We also propose architectures for all-optical circuits capable of performing the processing required for network coding. Our experiments show that network coding provides a moderate improvement in bandwidth efficiency for unprotected multicast while significantly outperforming existing approaches for dedicated multicast protection.

I. INTRODUCTION

O PTICAL crossconnect (OXC) technology for wavelength division multiplexing (WDM) optical networks is revolutionizing the infrastructure of many types of networks. A future where transparent optical networks dominate backbone communication networks is inevitable. Many of these OXC devices have the capability to support multicast connections at the physical layer, which has led to research into exploiting applications of optical-layer multipoint connections [2], [3], [4], [5], [6], [7], [8]. Not only will optical-layer multicast provide solutions for multicast applications such as high-bandwidth streaming multimedia and storage area networks, but it allows for the implementation of more general virtual topologies. These virtual topologies may be more bandwidth efficient and have an overall lower hardware cost [5].

Multicast communication has also been revolutionized with the advent of *network coding* in which logically disparate data is coded and transmitted together in order to obtain data rates not achievable with the traditional store-and-forward paradigm [9]. However, in order to apply network coding directly to optical networks, the signal must undergo optical-electricaloptical (OEO) conversion at each node so that the data can be buffered and processed for the coding operations. This opaque networking approach prohibits many of the benefits of opticallayer multicast. Furthermore, network coding is easier to apply in networks with fine bandwidth granularity. WDM networks, however, may be relatively coarse grained with individual communication channels operating in the tens of Gbps. In this paper, we address these issues and provide heuristics for coding on optical networks with coarse bandwidth granularity. We go over some related work in Section II, and in Section III, we discuss how network coding can be used with multicast-capable OXCs for a more transparent implementation. We provide our formulation for the problem of coding with coarse-grained bandwidth in Section IV followed by the heuristics for solving the problem in both the protected and unprotected cases in Section V. We then present our simulation results in Section VI. In Section VII, we present an architecture for an all-optical coding unit, and we conclude in Section VIII.

II. RELATED WORK

In this section, we review some related work on network coding as well as network coding applied to optical networks.

A. Network Coding

The concept of network coding was introduced in 2000 by Ahlswede, Cai, Li, and Yeung [9]. They showed that it is possible, using network coding, to achieve information rates in a single-source multicast equal to the smallest individual maxflow rates among the receivers. This represents a fundamental breakthrough as this rate is not achievable in traditional storeand-forward network models. Li, Yeung, and Cai later also showed that *linear network coding*, in which nodes transmit a linear combination of the data received on incoming links over each outgoing link, is sufficient to achieve the maximum flow rate [10].

The process of finding network codes for multicast can be broken into two sub-problems:

- 1) find a subgraph of the available topology which will be utilized in coding, and
- 2) find the algebraic operations which the nodes must perform to realize a valid code on the given topology.

Jaggi *et al.* presented both deterministic and randomized polynomial time algorithms for solving Subproblem 2 [11]. Lun *et al.* modeled Subproblem 1 as a linear program which finds the asymptotic optimal solution in polynomial time assuming arbitrarily fine bandwidth granularity [12]. In this paper, we look at Subproblem 1 in the context of coarse granularity.

B. Network Coding in Optical Networks

Much of the research on using network coding for optical networks relies on OEO conversion with electronic buffering and processing at each node. The use of photonic circuits for this purpose are only just beginning to be investigated [13].

A smaller, preliminary version of this work appeared in the Fifth International Conference on Broadband Communications (Broadnets 2008) [1].

E. Manley, J. Deogun, and L. Xu are with the Department of Computer Science and Engineering, University of Nebraska at Lincoln, Lincoln, Nebraska 68588-0115 USA.

D. Alexander is with the Department of Electrical Engineering, University of Nebraska at Lincoln, Lincoln, Nebraska 68588-0115 USA.

Bhattad *et al.* attempted to minimize the number of nodes at which coding is required for a multicast network [14]. An optical network then benefits by having fewer terminations and retransmissions of a signal when using optical circuit switching. With this kind of scheme there is a trade-off between the number of coding operations required, which necessitates terminating the optical signal, and bandwidth saved as a result of the coding.

Ahmed Kamal has employed coding as a mechanism for a novel protection scheme in optical networks called 1 + Nprotection [15], [16], [17], [18]. Under traditional shared protection in optical networks, backup bandwidth is provisioned for each connection and is only used in case of primary route failure. Several connections may share the same backup bandwidth if it can be assumed that their primary routes are unlikely to fail simultaneously. This is called 1: N protection because one unit of backup bandwidth protects N primary connections. One standard 1: N approach, *p*-Cycles, was introduced and extensively studied by Grover et al. [19], [20]. With p-Cycles, a bidirectional cycle of spare bandwidth is formed among several nodes in the network which can be used in case of a failure on the primary path between two end-nodes which are both on the cycle. Any links which are a chord to the cycle are also protected. However, by employing network coding on the *p*-Cycle, the protected connections simultaneously transmit their data (in coded form) on the backup bandwidth. In case of a primary failure, the end-nodes of the failed connection can recover the data from the coded backup transmission, and there is no need to detect the failure nor retransmit on the backup route which would hinder recovery time. However, the signal is terminated, processed, and retransmitted at each intermediate node on the backup route. The buffering and processing time then impacts the apparent recovery time, so it should be smaller than the detection and retransmission time of the alternative non-coding method. In the original work, Kamal presented a solution for the single-error model [15] and later extended it to handle multiple failures [16] and for GMPLS-based implementation [17]. A generalized version of the approach, which does not use p-Cycles was also introduced [18]. Kamal and Ramamoorthy proposed another generalization via implementation in an overlay layer [21]. It has also been proposed that the OEO conversion of this type of scheme could be avoided by utilizing a photonic XOR element [13]. We consider protected multicast in this paper as well as more general coding circuits.

III. CODING OPERATIONS IN OPTICAL NETWORKS

As defined by Yeung *et al.* [22], network codes can be described in terms of *local encoding mappings* for each node v,

$$\tilde{k}_e: F^{|\mathbf{In}(v)|} \to F$$

where F is the base field, In(v) is the set of channels coming in to v, Out(v) is the set of channels going out from v, and $e \in Out(v)$. This describes the computation that each node in the network must execute in order to implement the code but requires only local information. For linear network codes, \tilde{k}_e gives a linear combination of each symbol on the incoming



Fig. 1. An optical switch with a sharable bank of b optical circuits which perform coding operations.

channels. This is a relatively straightforward operation in electronic networks which can buffer packets and perform the computations with readily available ALU operations. If this process was implemented directly on an optical network, the signal would have to be terminated, converted to electronics, and buffered before computation and retransmission at each node. This conversion and buffering is undesirable for optical WDM mesh networks.

However, there exists optics technology that could enable these operations to a limited extent without OEO conversion or electronic buffering. All-optical buffers have also been proposed for use in certain optical communication applications such as Optical Packet Switching. These buffers are typically built using a series of fiber delay lines onto which the optical signal can be switched [23], but more sophisticated technology based on *slow-light* effects in optical microcavity resonators have also been investigated [24], [25]. With Optical Packet Switching, optical buffering is used to store a signal for a very short amount of time for the purpose of aligning the packet with an available outgoing time slot. A similar approach could be used for single-source network coding in optical networks. The symbols for coding could be delayed just long enough to be aligned before being input into an all-optical arithmetic unit which would perform a serial coding operation over the incoming signals.

Such an arithmetic unit could be built using all-optical logic gates and optics-on-a-chip technology [26], [27], [28], [29]. For instance, the simplest of codes could be supported by a single all-optical XOR gate built with an optical interferometer [28], [29]. The coarse bandwidth granularity forces all but a small number of the coefficients in the linear combination to be zero, so only a few incoming channels will factor into the computation which keeps the circuit complexity in check.

These operations devices could be deployed on a shared-perlink or shared-per-node basis at an all-optical cross-connect. For an illustration of such a switch layout, see Fig. 1. The details the all-optical serial arithmetic unit are presented in Section VII.

Network coding does have some disadvantages, and thus may not be desirable for every optical network. The processing of data required for network coding is suitable for electronic networks because they are opaque. While complete transparency is not possible for network coding in optical networks, coding operations may be handled in a way that maintains many of the advantages of transparency. The switches must be aware of information such as data representation and data rates. Furthermore, the control scheme must take into account varying propagation delays which impacts the length of the optical buffering. Since semi-permanent streams of traffic will minimize the transient effects of propagation delays, bursty traffic may not necessarily benefit from network coding. In order to complexities arising from varying data rates, in this paper, we only address single-source coding.

IV. PROBLEM FORMULATION

As discussed in Section II, the coding problem can be split into two independent subproblems.

Given a particular multicast request of rate r between a source s and a set of sinks T, there exists a solution to Subproblem 2 if there exists a solution to Subproblem 1 which contains a flow of size r between s and t for each $t \in T$. Note that for sinks $t_1, t_2 \in T$, the flows to t_1 and t_2 respectively need not be independent. Yeung *et al.* provide a polynomialtime algorithm for Subproblem 2 which codes over symbols of size $O(\log |T|)$ [22]. Thus, the problem of finding efficient network coded routes is reduced to Subproblem 1.

For general multicast networks, Lun *et al.* formulated Subproblem 1 as a linear program which finds an asymptotically optimal solution in polynomial-time [12]. This method is practical for electronic packet-based networks which can split a message into arbitrary fractions across different communication channels. However, with a coarse bandwidth granularity, this method will not necessarily produce usable solutions. In optical networks, the bandwidth granularity is usually extremely coarse, often with a minimum subdivision on the order of tens of Gbps. Thus, for our purposes, it is more appropriate to model the problem as follows.

Minimum Cost Coded Flow Problem:

Instance: A directed weighted multigraph G = (V, E), a source vertex $s \in V$, a set of sink vertices $T \subseteq V$ with $s \notin T$, and a positive integer r.

Question: Find a minimum cost subgraph G' = (V', E') of G with $T \cup \{s\} \subseteq V'$ such that for each $t \in T$, G' contains r edge-disjoint paths between s and t.

In this model, each edge represents the minimum subdivision of bandwidth (e.g. one edge could be equivalent to one wavelength on a fiber).

In fact, the directed Steiner Tree problem can be reduced to the Minimum Cost Coded Flow problem for each $r \in \mathbb{N}$,



and the directed Steiner Tree problem is known to be NP-hard [30]. We sketch here the proof of this for r = 2. It should be easy to see how this generalizes for all $r \in \mathbb{N}$.

Suppose G_{DST} is a graph on which we wish to find a directed Steiner tree from the source s to sink set T. We can transform this into an instance of Minimum Cost Coded Flow by creating 2 copies of G_{DST} : G_{DST}^1 and G_{DST}^2 with $G_{DST} \cong G_{DST}^1 \cong G_{DST}^2$. Let s^1 and s^2 be the vertices corresponding to s in G_{DST}^1 and G_{DST}^2 respectively. Similarly, let T_1 and T_2 be the sink sets corresponding to T. Create a new source s' with an edge to each of s^1 and s^2 . Create a new sink set T' which contains one new vertex t'_i for each $t_i \in T$. Then from each $t_i^1 \in T_1$, create an edge from t_i^1 to t'_i . Similarly, create edges from t^2_i to t'_i . Now consider a solution to the transformed problem and suppose $t' \in T$. Given that t' has exactly one incoming edge from each of G^1_{DST} and G^2_{DST} , exactly one path from s' to t' is in each of \tilde{G}_{DST}^1 and \tilde{G}_{DST}^2 . Therefore, the portion of the subgraph contained in G_{DST}^1 gives a minimum Steiner tree for G_{DST} . Conversely, 2 copies of the minimum Steiner tree gives a minimum solution to the transformed instance of the Minimum Cost Coded Flow problem. For an illustration of the reduction, see Fig. 2. Therefore, for any $r \in \mathbb{N}$, the Minimum Cost Coded Flow Problem is NP-hard.

Kiraly and Lau formulated a similar problem in which the topology is undirected. The problem asks for a minimum cost orientation (cost varies depending on which direction an edge is oriented) of the graph such that the source is r-arc-connected to each sink. This problem was shown to be NP-Complete and hard to approximate within a factor of log [31], [32]. This



problem generalizes to an oriented hypergraph problem which was motivated by the coded multicast problem in wireless networks in which the tail of a hyperedge is the transceiver and the head is the set of receivers of a wireless transmission.

Note that this problem statement considers the cost of bandwidth only and does not require wavelength continuity. We leave the problem of optimizing over additional costs (e.g. the cost of the optical circuitry for implementing the coding operations) and the incorporation of wavelength assignment as future work.

V. HEURISTICS

In this section, we present two heuristics for solving the Minimum Cost Coded Flow problem for optical WDM networks.

This problem is difficult because the minimum cost coded multicast may simply be multiple copies of a minimum directed Steiner tree. In this instance, it makes sense to simply compute a good Steiner tree using an approximation algorithm and allocate multiple wavelengths along the resulting tree. Consider the two-wavelength butterfly network with unit edge cost shown in Fig. 3. When s = A, $T = \{F, G\}$, r = 2, the optimal solution using 8 edges is achieved by a two-unit Steiner tree as shown in Fig. 3(a). Now suppose D is added to the sink set ($T = \{D, F, G\}$). We might expect a similar solution as shown in Fig. 3(b). However, this solution uses 10 edges whereas the network coded solution in Fig. 3(c) uses 9. While a 10% bandwidth savings here may not seem significant, the throughput doubles, in this case, when network coding is used.

A. Multicast with Network Coding

Our approach is motivated by the classic shortest path heuristic [33] for finding Steiner trees in undirected networks. In this algorithm, a tree is built iteratively by selecting the path which will result in the least-cost increase to the partial solution. We generalize this concept by applying it in a rooted, directed fashion for r-edge-connectivity. Two variations are presented here in Algorithms 1: Coded Multicast using Arbitrary node ordering (CMA) and 2: Coded Multicast using Greedy node ordering (CMG).

1) CMA Heursitic: In CMA the subgraph is built iteratively, starting with a Steiner 1-edge-connected subgraph which then increments to a 2-edge-connected graph and so on. At each iteration, we consider the nodes in an arbitrary order, finding the path to each node which is disjoint from all other paths to that node found in previous iterations but which causes the smallest increase in cost to the subgraph.

CMA has complexity O(r|T|(|V| + |E|)). To see this, note that each edge in the current portion of the subgraph H_0 is examined exactly once between lines 7 and 12 (either for deletion of setting of zero weight in G_0) because each of $p_{v,1}, p_{v,2}, \ldots, p_{v,i-1}$ are edge disjoint. It is also evident that each of lines 5, 6, 13 (using some linear-time shortest path algorithm), and 14 take O(|V| + |E|). This block must be executed r|T| times. This dominates theO(|V|), O(r|T|), and O(1) running times of lines 1, 2, and 17, so the time



Fig. 3. Multicast sessions in the butterfly topology with two wavelengths of unit edge cost. Allocated bandwidth is represented by heavier, dashed lines.

complexity of the algorithm is O(r|T|(|V| + |E|)). With coarse-grained networks, r is small enough that is can be considered constant. Growth in r would correspond to a higher degree of stream divisibility, and therefore, finer granularity with respect to the total data rate of the multicast session.

2) CMG Heursitic: CMG is similar to CMA except that at each iteration, the subgraph is incremented using a greedy ordering. To that end, a new potential path is computed for each sink, and only the path which results in the smallest increase in cost to the subgraph H is made permanent. This results in lines corresponding to 5 through 13 of CMA being run $O(r|T|^2)$ times instead of O(r|T|) resulting in a total running time of $O(r|T|^2(|V| + |E|))$.

B. Protected Multicast with Static Network Coding

These heuristics can be easily altered to find subtopologies for multicast sessions which are robust against link failures using *static network codes*. Static network codes are codes which are robust against a particular set of potential linkfailure patterns. If one of these given failure patterns occurs, Algorithm 1 CMA: Coded Multicast using an Arbitrary node ordering

- **Input:** directed multigraph G, source s, sink set T, r units of bandwidth requested
- **Returns:** subgraph *H* containing *r* disjoint paths between *s* and *v* for each $v \in T$
- 1: Initialize subgraph $H \leftarrow (V, \emptyset)$.
- 2: Create an initially empty $|T| \times r$ path matrix p such that $p_{v,i}$ will store the *i*th edge disjoint path from s to v.
- 3: for i from 1 to r do
- 4: for $v \in T$ do
- 5: Set $G_0 \Leftarrow G$
- 6: Set $H_0 \Leftarrow H$
- 7: **for** *j* from 1 to i 1 **do**
- 8: Delete edges on $p_{v,j}$ from both G_0 and H_0 .
- 9: **end for**
- 10: for each remaining edge $e \in H_0$ do
- 11: Set the weight of e in G_0 to 0.
- 12: **end for**
- 13: Find $p_{v,i}$, the shortest *s*-*v* path in G_0 .
- 14: Set $H \leftarrow H \cup p_{v,i}$.
- 15: **end for**
- 16: end for
- 17: **return** *H*

then the message can be recovered at the sinks without changing any of the local encoding mappings at any of the network nodes. In optical networking, this serves the same function as dedicated protection of a route, in which two copies of a message are sent: one on the primary route, and the other on the backup route. In case of a link failure, the receiver for the protected connection may simply reconfigure to accept the message on the backup route. With both static network codes and optical-layer dedicated multicast protection, no retransmission is needed, and no crossconnect needs to detect the failure or reconfigure to recover from the failure.

Koetter and Médard [34] proved that a static network code exists for a multicast session in a topology G which is robust against a set F of failure patterns (such that each $f \in F$ is a set of edges defining the failure) if there exists a network code for the multicast in the topology G - f for all $f \in F$.

Using this result, we obtain robust versions of CMA and CMG, which we refer to as RCMA and RCMG respectively, by replacing line 8 in CMA and line 13 in CMG with the following line:

```
for each edge (x, y) on p_{v,j}, delete all edges
with endpoints x and y from both G_0 and H_0
```

which will ensure that for each $t \in T$, the *r* paths from *s* to *t* are link-disjoint rather than simply edge-disjoint. This will protect a single-unit of multicast bandwidth against a single-link failure by finding a subgraph which is Steiner 2 link-connected and computing its static network code.

Given a suitable subgraph for protecting against a singlelink failure, a static network code can be found in polynomial time over a field containing $2^m(m+1)$ elements where m is the total number of channels in the subgraph [22]. Algorithm 2 CMG: Coded Multicast using a Greedy node ordering

- **Input:** directed multigraph G, source s, sink set T, r units of bandwidth requested
- **Returns:** subgraph *H* containing *r* disjoint paths between *s* and *v* for each $v \in T$
- 1: Initialize subgraph $H \leftarrow (V, \emptyset)$.
- 2: Create an initially empty $|T| \times r$ path matrix p such that $p_{v,i}$ will store the i^{th} edge disjoint path from s to v.
- 3: for i from 1 to r do
- 4: Set $T_0 \leftarrow T$.
- 5: while $T_0 \neq \emptyset$ do
- 6: Initialize node $v_{best} \leftarrow \textbf{null}$.
- 7: Initialize path $p_{best} \leftarrow$ **null**.
- 8: Initialize cost $c_{best} \leftarrow \infty$.
- 9: for $v \in T_0$ do
- 10: Set $G_0 \Leftarrow G$

11:

12:

13:

16:

19:

20:

21:

22:

- Set $H_0 \Leftarrow H$
- for j from 1 to i-1 do
 - Delete edges on $p_{v,j}$ from both G_0 and H_0 .
- 14:end for15:for each remaining edge $e \in H_0$ do
 - Set the weight of e in G_0 to 0.
 - end for
- 17: end for 18: Find $p_{v,i}$
 - Find $p_{v,i}$ with cost c, the shortest s-v path in G_0 .

```
if c < c_{best} then
```

```
\begin{array}{l} \textbf{Set } c_{best} \Leftarrow c \\ \textbf{Set } p_{best} \Leftarrow p_{v,i} \\ \textbf{Set } v_{best} \Leftarrow v \end{array}
```

- 23: end if
- 24: end for
- 25: Set $H \leftarrow H \cup p_{best}$.
- 26: Set $T_0 \leftarrow T_0 v_{best}$.
- 27: end while
- 28: end for
- 29: **return** *H*

Without coding, in order to guarantee dedicated protection against the failure of a single link, a backup tree must be computed which is link disjoint from the primary tree. Another approach for the protection of optical layer multicast is the idea of self-sharing trees which allow certain portions of the backup bandwidth to protect against different link failures on the primary tree [35]. The primary and backup bandwidth in this approach must collectively contain two link-disjoint paths between the source and each sink in order to protect against a single failure, so collectively, this bandwidth forms a subtopology suitable for dedicated protected multicast with network coding (assuming edges are directed rather than orientable). Therefore, in terms of bandwidth, allowing coding is no less efficient than using self-sharing but has the added benefit of being dedicated protection rather than shared protection. However, cross-sharing, i.e. sharing backup bandwidth between two different multicast sessions [36], may allow for some bandwidth savings over the coded approach, but contrary



Fig. 4. Topologies used in the simulations.

to the coding approach, will suffer delay from reconfiguration and retransmission.

Intuitively, the reason that network coding has the potential to greatly improve upon the dedicated backup tree method is that the network coding approach guarantees protection for the whole group given a local connectivity requirement between the source and each sink. The backup tree method requires a much more stringent global connectivity property in order to protect the whole group. A route satisfying this property is more difficult to find and less likely to exist.

VI. SIMULATION RESULTS

We conducted simulations in order to determine the potential benefit of using network coding for both optical layer multicast as well as protected optical layer multicast. For both cases, we use a network model in which each link in the network contains enough unused bandwidth to support all requests. Comparisons are then made with non-network coding solutions.



(b)

fraction of nodes in the multicast group

Fig. 5. For the Pacific Bell network under CMA; (a) the fraction of the multicast sessions in our simulation which benefit from network coding, (b) the fraction of bandwidth saved for those sessions in which coding was beneficial.

In this section, we present the results on three topologies (shown in Figure 4): the 15-node Pacific Bell network, the 21node Italian network [37, p. 193], and a randomly generated 50-node network. The random network was generated using the rectangular grid method [38]. Each data point presented is the average over 1000 different multicast sessions. The source for each simulated multicast session was chosen randomly from a uniform distribution. A fixed number of sinks was then also chosen uniformly at random.

A. Multicast

For the case of unprotected multicast, we simulated both the CMA and CMG heuristics. We compared each algorithm against the traditional multicast algorithm obtained from running multiple instances of the heuristic with only one unit of bandwidth requested (which will, by definition not require any network coding). That is, we simply compute a single-unit multicast session, remove the allocated edges, and then repeat until we have allocated the desired amount of bandwidth for the r-unit multicast. Since all links have enough available bandwidth to support all requests in our model, this is effectively r copies of the same directed Steiner tree. This comparison allows us to see the impact that network coding has vis-a-vis a non-network coding algorithm of the same basic approach and complexity. Thus, we can attribute



Pacific Bell Network - Greedy Order

(a)





(b)

Pacific Bell Network - Greedy Order

Italian Network - Arbitrary Order



fraction of nodes in the multicast group

(a)

Italian Network - Arbitrary Order



Fig. 6. For the Pacific Bell network under CMG; (a) the fraction of the multicast sessions in our simulation which benefit from network coding, (b) the fraction of bandwidth saved for those sessions in which coding was beneficial.

any improvement over traditional multicast to network coding rather than tertiary differences between algorithms.

Recall that in many cases, there is no improvement possible for a particular multicast session [see Fig. 3(a)]. In fact, in the optimal case, the improvement in throughput due to coding is conjectured to be bounded by a factor of two [39]. In practice, we found no advantage at all to using coding (even with the optimal subgraph) when compared with an optimal multicast route. A similar observation was also made by Li, Li, and Lau:

"the fundamental benefit of network coding is not higher optimal throughput, but to facilitate significantly more efficient computation and implementation of strategies to achieve such optimal throughput." [39]

This is the reason why we chose to study the performance of each heuristic against its corresponding traditional multicast algorithm rather than the optimal case. However, unlike Li, Li, and Lau, we consider a fixed desired throughput and then measure bandwidth usage. In order to better understand the impact of the cases in which coding does offer a more efficient solution, we first look at the fraction of multicast sessions in which network coding allowed a nonzero improvement, and then for those cases, we computed the amount of bandwidth that was saved as a fraction of the bandwidth required using the traditional multicast approach. These values were for

various multicast group sizes and bandwidth request amounts. The results are plotted in Figures 5 through 10.

Fig. 7. For the Italian network under CMA; (a) the fraction of the multicast

sessions in our simulation which benefit from network coding, (b) the fraction

of bandwidth saved for those sessions in which coding was beneficial.

We found that, for both algorithms on all topologies the fraction of sessions which benefit from coding was lowest when the multicast group size was very small or very large and peaked somewhere in the middle. This seems to be consistent with theoretical results which state that there is no advantage to network coding in the case of either unicast or broadcast [40]. However, the fraction of beneficial coding sessions with the arbitrary-order heuristic peaks with a relatively larger group size. The limited sample of network topologies also seems to indicate that larger networks have a greater potential for coding benefit. This is again likely because, with larger networks, there are many more potential solutions, and thus the nearoptimal ones are more difficult to isolate.

Also, the fraction of comparatively beneficial sessions seems to be an order of magnitude higher for CMA. This is because the greedy-order heuristic for the uncoded traditional multicast variation finds many more optimal or nearoptimal routes which are very difficult to improve upon as stated earlier. This tends to suggest that network coding and higher-complexity heuristics are two options for improving the bandwidth efficiency of multicast. We leave the details of this hardware space/time complexity trade-off to future work.

These simulations also show that the amount of bandwidth



bandwidth

requested

Fig. 8. For the Italian network under CMG; (a) the fraction of the multicast sessions in our simulation which benefit from network coding, (b) the fraction of bandwidth saved for those sessions in which coding was beneficial.

(b)

0.40

0.50

0.60 0.70

fraction of nodes in the multicast group

0.80

6 0.0

requested has very little impact on the fraction of sessions which see an improvement. It may be that these requests simply are too-coarse to show this impact. However, the bandwidth savings when coding does give an improvement increases with the amount of bandwidth requested which indicates that finer-grained bandwidth subdivision relative to the total amount of bandwidth requested does indeed allow for more bandwidth efficient solutions.

One surprising finding, however, is that the amount of bandwidth saved seems to be similar regardless of the algorithm or topology. In all cases, our simulation showed a general decrease in savings as group size increased and an increase in savings as the requested bandwidth increases. Again, this is likely due in part to the granularity and near-optimal strength of the base algorithms as described above. However, it also suggests that there may be typical route sub-topologies that occur frequently and are improved upon in the same way regardless of the algorithm and larger network topology. For instance, variations of the butterfly example (see Fig. 3) may occur often. But, when the two or three sinks involved in the "butterfly" part of the route become an increasingly smaller part of the entire coding subgraph, the impact of the improvement on that part also become relatively smaller.

50-Node Network - Arbitrary Order



fraction of nodes in the multicast group

50-Node Network - Arbitrary Order



Fig. 9. For the random 50-node network under CMA: (a) the fraction of the multicast sessions in our simulation which benefit from network coding, (b) the fraction of bandwidth saved for those sessions in which coding was beneficial.

B. Protected Multicast

We also simulated RCMG, the greedy-order variation of our heuristic, for robust multicast. as indicated in Section V-B with r = 2 to get a subgraph with two link-disjoint paths between the source and each sink. If the source is 2 link-connected to each sink and the shortest-path algorithm used in line 18 is altered to instead find the shortest path which does not block all other s - v paths (e.g. run Suurballe's algorithm [41] for finding the shortest path pair and use only the shorter of the two paths), then this algorithm is guaranteed to find a solution. This is a tremendous advantage over traditional optical-layer multicast dedicated protection algorithms which often fail to find two disjoint directed Steiner trees.

For comparison with other optical-layer multicast protection algorithms, we implemented a naive algorithm which attempts to find two link-disjoint directed Steiner trees by first using CMG with r = 1, deleting all links containing edges in the resulting tree, and then running CMG with r = 1 a second time to find a backup tree.

We also implemented a heuristic called the Minimum Cost Collapsed Ring (MCCR) algorithm which was recently proposed by Rahman and Ellinas [42] for dedicated multicast protection and reported to have low blocking. This approach uses the inherent robustness of the ring structure to construct

50-Node Network - Greedy Order fraction of sessions benefitted by coding 0.3 0.2 0.1 units of 0 bandwidth 0.04 0.14 0.24 0.35 requested 0.45 0.55 0.65 0.76 0.86 0.96 fraction of nodes in the multicast group (a)



Fig. 10. For the random 50-node network under CMG; (a) the fraction of the multicast sessions in our simulation which benefit from network coding, (b) the fraction of bandwidth saved for those sessions in which coding was beneficial.

a protected multicast. First, a ring is found, and two paths are set up in opposite directions on different wavelengths which each terminate at the last node before the source.

We experimented with all three algorithms on our three test networks and measured the blocking rate as well as the bandwidth used by each session. The blocking rate is defined as the number of requested sessions which can not be fulfilled over the total number of requested sessions. Note that since we assume that all links have enough available bandwidth to support all connections, a connection cannot be blocked because some links are saturated due to inefficient routing of previous sessions. Thus, all blocking is due to the fact that the topology does not admit a solution (a case in which the network coding approach has an advantage) or because a solution exists and the algorithm fails to find it. The blocking rates of the three algorithms are shown over varying multicast group sizes in Figures 11(a), 12(a), and 13(a).

After comparing the blocking rates, we consider bandwidth efficiency by comparing each of the naive and MCCR algorithms independently against the network coding approach. Since a different set of sessions are blocked by each algorithm, we only look at the subset of connections in which both the naive algorithm and the coding heuristic (respectively the MCCR algorithm and the coding heuristic) found a solution. In



Fig. 11. For the Pacific Bell network; (a) the algorithm blocking rate for each of the naive heuristic, MCCR heuristic, and our RCMG coding heuristic, (b) the amount of additional bandwidth needed by the naive heuristic (resp. MCCR heuristic) for those sessions in which both the naive and coding heuristics (resp. MCCR and coding heuristics) found a valid solution.

our simulations, no session was ever blocked using the network coding heuristic, so these comparisons do not overstate the relative bandwidth efficiency of the coding heuristic. The data points in those plots [Figures 11(b), 12(b), and 13(b)] is the percent of additional bandwidth needed when using the traditional protected multicast approach versus the network coding approach. That is, we plot

$$\frac{\beta_{\rm nc} - \beta_{\rm c}}{\beta_{\rm c}} \times 100$$

where β_{nc} is the bandwidth used with non-coding approach and β_{c} is bandwidth used with coding approach

In the Pacific Bell network [Figure 11(a)], the MCCR algorithm blocked significantly fewer sessions than the naive algorithm steadily increased blocking as the size of the groups grew larger. However, the network coding approach outperformed both algorithms, not blocking any of the sessions. For those sessions in which the naive algorithm did not block, it used up to 10% more bandwidth than the coding algorithm [Figure 11(b)]. Even though the MCCR algorithm blocked less often than the naive algorithm, it used relatively more bandwidth compared with the coding approach, using as much as 35% more.

For the two larger networks [Figures 12(a) and 13(a)], the







(b)

Random 50-Node Network



(b)

fraction of nodes in the multicast group

Fig. 12. For the Italian network; (a) the algorithm blocking rate for each of the naive heuristic, MCCR heuristic, and our RCMG coding heuristic, (b) the amount of additional bandwidth needed by the naive heuristic (resp. MCCR heuristic) for those sessions in which both the naive and coding heuristics (resp. MCCR and coding heuristics) found a valid solution.

Fig. 13. For the random 50-node network; (a) the algorithm blocking rate for each of the naive heuristic, MCCR heuristic, and our RCMG coding heuristic, (b) the amount of additional bandwidth needed by the naive heuristic (resp. MCCR heuristic) for those sessions in which both the naive and coding heuristics (resp. MCCR and coding heuristics) found a valid solution.

MCCR outperformed the naive algorithm, but to a lesser extent than with the Pacific Bell network. In both these instances the blocking rate still approached 1.0 with the increase in group size. The trend in additional bandwidth used also continued [Figures 12(b) and 13(b)] with the naive algorithm using up to 10% more bandwidth. However, the MCCR algorithm used as much as 50% more bandwidth in the Italian network and as much as 70% more in the random 50-node network.

Given these results, it appears that network coding provides a very good solution for robust multicast in optical networks as it has significantly lower blocking and uses less bandwidth than existing approaches.

VII. ARCHITECTURE OF AN ALL-OPTICAL CODING UNIT

Now that we have established potential benefits of network coding in optical networks, we look at how the coding operations might be implemented. In this section, we give an architecture capable of performing the coding arithmetic in $GF(2^m)$, the finite field with 2^m elements. Symbols in $GF(2^m)$ can be represented using m bits where each bit represents a coefficient to the polynomial $x^{m-1} + x^{m-2} + \cdots + x^2 + x + 1$. Addition is accomplished using a bit-wise XOR, and multiplication is done modulo some irreducible polynomial of degree m. In $GF(2^m)$, a linear multicast code may be constructed in polynomial time supporting $2^m - 1$ terminals using an algorithm of Jaggi *et al.* [11], [22]. We denote the physical bit separation as τ . An overview of the architecture is shown in Fig. 14. Given two serial inputs A and $B \in GF(2^m)$, the coding unit first performs the scalar multiplication and addition $C_A \cdot A + C_B \cdot B$ using the coefficients C_A and C_B from the local encoding kernal. This multiplication takes 2m bits on the serial line, so two of these units run in parallel in order to avoid the need for buffering. The stream of input symbols are multiplexed between these two units. This can be accomplished using a high-speed 1×2 switch such as the one demonstrated by Herrera *et al.* which operates at 160 Gbps [43]. The resulting 2m-bit values are normalized to *m*-bit values concurrently using the normalization unit. We will now describe the scalar multiplication/addition unit and the normalization unit.

A. Scalar Multiplication/Addition Unit

Multiplication in this architecture is accomplished using the shift-and-add approach. The input symbols are split into m copies and then fiber delay lines shift each copy by the appropriate number of bits. The fiber delay lines could be set for a fixed delay or could be controllable to accommodate multiple transmission rates. Each bit of the scalar coefficient controls an all-optical "ON/OFF" switch, such as a semiconductor optical amplifier (SOA), which effectively multiplies the correspond-



Fig. 14. An overview of the architecture for an all-optical coding unit.



Fig. 15. An all-optical circuit for scalar multiplication and addition in $GF(2^m)$.

ing copy of the input signal by that bit. The coefficient-bit control signal is fixed for a particular session, so high speed switching is not necessary here. Addition in $GF(2^m)$ is merely an XOR of the operands, and can be accomplished with an all-optical XOR gate [27], [28], [29]. Figure 15 shows a diagram of this all-optical circuit. This design uses two 1 : N optical splitters, 2m - 2 delay lines (which need not allow the same maximum delay), 2m SOAs, and at most m two-input all-optical XOR gates.

B. Normalization Unit

The normalization unit in Fig. 16 shows how two 2m-1-bit outputs of the multiplication/addition units can be normalized with a reducing polynomial of the form $x^m + x + 1$. This polynomial is irreducible for several choices of m (such as m = 2, 3, 4, 6, 7, 9, 15, 22, 28, 30, 46, 60 and 127 [44, p. 158]). If another choice of m is desired, the architecture can be easily modified for an appropriate modulus.

Our design is based on the GF(2^m) normalization step in Itoh-Tsujii algorithm [45], [46]. This approach reduces a polynomial of the form $d_{2m-2}x^{2m-2} + d_{2m-3}x^{2m-3} + d_{2m-4}x^{2m-4} + \ldots + d_3x^3 + d_2x^2 + d_1x + d_0$ where $d_i \in GF(2)$ by noticing that

That is, the term $d_i x^i$ can be removed for all $i \ge m$ and replaced with two terms in $GF(2^m)$. Thus, the polynomial becomes $r_{m-1}x^{m-1}+r_{m-2}x^{m-2}+r_{m-3}x^{m-3}+\ldots+r_3x^3+r_2x^2+r_1x+r_0$ where

$$\begin{array}{rcrcrcrcrcrc} r_{0} & = & d_{0} & \oplus & d_{m} \\ r_{1} & = & d_{1} & \oplus & d_{m+1} & \oplus & d_{m} \\ r_{2} & = & d_{2} & \oplus & d_{m+2} & \oplus & d_{m+1} \\ r_{3} & = & d_{3} & \oplus & d_{m+3} & \oplus & d_{m+2} \\ & & \vdots & & & \\ r_{m-3} & = & d_{m-3} & \oplus & d_{2m-3} & \oplus & d_{2m-4} \\ r_{m-2} & = & d_{m-2} & \oplus & d_{2m-2} & \oplus & d_{2m-3} \\ r_{m-1} & = & d_{m-1} & \oplus & & d_{2m-2}. \end{array}$$



Fig. 16. A serial all-optical circuit for normalization in $GF(2^m)$ when the reducing polynomial is $x^m + x + 1$ (Note: This is not a valid for m in which $x^m + x + 1$ is reducible).



Fig. 17. An illustration of using a parallel normalization circuit with the appropriate serial-to-parallel and parallel-to-serial conversion.

Thus, we may XOR the lower m bits with two copies of the upper m-1 bits (with the second copy being up-shifted one bit). This operation is accomplished by splitting the symbol onto two space-disjoint lines using an $m\tau$ clock as a control signal and delaying the lower bits by $m\tau$ in order to align them with the upper bits. The upper bits are split into two copies with a one-bit delay on one copy. The three signals can then be XORed to get the normalized value.

In the portion of the circuit where the value is split onto three separate lines, the value takes only *m*-bits of space, so this is a prudent location to multiplex the two outputs from the multiplication/addition unit as illustrated in Fig. 16. This design requires two 1×2 switches, a 1:2 splitter, two 2:1 combiners, two fiber delay lines, and two two-input all-optical XOR gates.

Alternatively, normalization in $GF(2^m)$ may take place using a parallel normalization unit (see Fig. 17). This approach has been proposed for inband-forward-error-correction in SDH/SONET networks [47]. This would also require the use of serial-to-parallel and parellel-to-serial converters. This technology has been proposed for use in label recognition [48], [49], [50]. A design for the parallel normalization unit using the $x^m + x + 1$ modulus is shown in Fig. 18 which uses 2m - 1 two-input XOR gates as well as m splitters. In this design, only fixed delay lines are needed to ensure the proper propagation delay on each of the inputs.

VIII. CONCLUSIONS

In this paper, we investigated several issues related to the use of network coding for optical-layer multicast. We formulated the problem of finding a subgraph of the optical topology for coding and discussed its complexity. We gave heuristics for finding bandwidth efficient subgraphs for use in coding



Fig. 18. A parallel all-optical circuit for normalization in $GF(2^m)$ when the reducing polynomial is $x^m + x + 1$. This design is easily modifiable for different reducing polynomials.

for both protected and unprotected multicast. Additionally, we have proposed architectures for supporting coding operations in optical hardware while maintaining as much transparency as possible.

Our simulation results have shown that network coding can improve the bandwidth efficiency of mutlicast sessions over analogous directed Steiner tree heuristic methods even in optical networks which have relatively coarse-grained subdivisions of bandwidth. The fraction of multicast groups which could be improved varies depending on the topology and the heuristic used, but the overall trends are similar. In general, network coding has the greatest potential benefit in the less-complex algorithms where coding provides an option for increased efficiency as opposed to algorithms of greater complexity which find better routes for non-coded multicast. Furthermore, we showed that the network coding approach to dedicated protection is far superior than existing methods in terms of blocking rate and bandwidth usage.

Some future research directions include investigating the problem of coding over multiple sessions from multiple sources and exploring the bandwidth/hardware trade off. There is also theoretical work to be completed related to network coding in coarse-grained networks. Some of these questions are well-known including a conjecture of Kriesell's [51] and a question of Li and Li on the theoretical coding advantage using coding in undirected networks with integral routing [40].

REFERENCES

- E. D. Manley, J. S. Deogun, and L. Xu, "Network coding for opticallayer multicast," in *Proceedings of the Fifth International Conference* on Broadband Communications (Broadnets), September 2008.
- [2] M. Ali and J. S. Deogun, "Allocation of splitting nodes in wavelengthrouted networks," *Photonic Net. Commun.*, vol. 2, no. 3, pp. 245–263, August 2000.
- [3] —, "Power-efficient design of multicast wavelength-routed networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1852–1862, October 2000.
- [4] H. S. Hamza and J. S. Deogun, "A new class of strictly nonblocking WDM multicast switching networks," in *Proceedings of the Forth International Conference on Broadband Communications (Broadnets)*, 2007.
- [5] G. Rouskas, "Optical layer multicast: rationale, building blocks, and challenges," *IEEE Netw.*, vol. 17, no. 1, pp. 60–65, Jan/Feb 2003.
- [6] L. Sahasrabuddhe and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Commun. Mag.*, vol. 37, no. 2, pp. 67–73, Feb 1999.
- [7] D. Thaker and G. N. Rouskas, "Multi-destination communication in broadcast WDM networks: A survey," *Optical Networks Magazine*, vol. 3, no. 1, pp. 34–44, January/February 2002.
- [8] X. Wang, S. Wang, L. Li, and S. Xu, "Achieving shared multicast protection in wdm networks with sparse-light-splitting constraint," *J. Opt. Netw.*, vol. 7, no. 1, pp. 1–14, 2008.
- [9] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
- [10] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371 381, February 2003.
- [11] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, June 2005.
- [12] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom*, vol. 3, March 2005, pp. 1607–1617.
- [13] R. Menendez and J. Gannet, "Efficient, fault-tolerant all-optical multicast networks via network coding," in *Conference on Optical Fiber communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*, February 2008, pp. 1–3.

- [14] K. Bhattad, N. Ratnakar, R. Koetter, and K. Narayanan, "Minimal network coding for multicast," in *Proc. International Symposium on Information Theory (ISIT)*, September 2005, pp. 1730–1734.
- [15] A. E. Kamal, "1+N protection in optical mesh networks using network coding on p-cycles," in *Proceedings of IEEE Globecom*, 2006.
- [16] —, "1+N protection against multiple link failures in mesh networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Glasgow, Scotland, UK, June 2007.
- [17] ——, "GMPLS-based hybrid 1+N link protection over p-cycles: Design and performance," in *Proceedings of IEEE Globecom*, 2007.
- [18] —, "A generalized strategy for 1+N protection," in Proceedings of the IEEE International Conference on Communications (ICC2008), 2008.
- [19] D. Stamatelakis and W. D. Grover, "Theoretical underpinnings for the efficiency of restorable networks using pre-configured cycles ("pcycles")," *IEEE Trans. Commun.*, vol. 48, no. 8, pp. 1262–1265, 2000.
- [20] A. Kodian and W. D. Grover, "Failure independent path-protecting pcycles: efficient and simple fully pre-connected optical-path protection," *J. Lightw. Technol.*, vol. 23, no. 10, October 2005.
- [21] A. E. Kamal and A. Ramamoorthy, "Overlay protection against link failures using network coding," in *Proceedings of the Conference on Information Sciences and Systems (CISS)*, 2008.
- [22] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*. Hanover, Massachusetts, USA: now Publishers Inc., 2006.
- [23] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *J. Lightw. Technol.*, vol. 16, no. 12, pp. 2081–2094, December 1998.
- [24] P. Ku, C. Chang-Hasnain, and S. Chuang, "variable semiconductor alloptical buffer," *Elect. Lett.*, vol. 38, pp. 1581–1583, November 2002.
- [25] C. Chang-Hasnain, P.-C. Ku, J. Kim, and S.-L. Chuang, "Variable optical buffer using slow light in semiconductor nanostructures," *Proc. IEEE*, vol. 91, no. 11, pp. 1884–1897, November 2003.
- [26] A. Sharaiha, H. Li, F. Marchese, and J. Le Bihan, "All-optical logic nor gate using a semiconductor laser amplifier," *Elect. Lett.*, vol. 33, no. 4, pp. 323–325, 13 Feb 1997.
- [27] G. Theophilopoulos, K. Yiannopoulos, M. Kalyvas, C. Bintjas, G. Kalogerakis, H. Avramopoulos, L. Occhi, L. Schares, G. Guekos, S. Hansmann, and R. Dall'Ara, "40 ghz all-optical xor with uni gate," in *Optical Fiber Communication Conference and Exhibit (OFC)*, vol. 1, 2001, pp. MB2–1–MB2–3 vol.1.
- [28] Q. Wang, G. Zhu, H. Chen, J. Jaques, J. Leuthold, A. Piccirilli, and N. Dutta, "Study of all-optical xor using mach-zehnder interferometer and differential scheme," *IEEE J. Quantum Electron.*, vol. 40, no. 6, pp. 703–710, June 2004.
- [29] R. Webb, R. Manning, G. Maxwell, and A. Poustie, "40 gbit/s all-optical XOR gate based on hybrid-integrated mach-zehnder interferometer," *Electronics Letters*, vol. 39, no. 1, pp. 79–81, January 2003.
- [30] M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed steiner problems," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998, pp. 192–200.
- [31] T. Király and L. C. Lau, "Approximate min-max theorems for steiner rooted-orientations of graphs and hypergraphs," *Journal of Combinatorial Theory, Series BIn Press*, vol. Proof, March 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6WHT-4S3G9JM-1/2/555dfee6206e1b7e13c1f308038b073d
- [32] T. Király and L. L. C., "Approximate min-max theorems for steiner rooted-orientations of graphs and hypergraphs," Egerváry Research Group, Budapest, Tech. Rep. TR-2006-13, 2006.
- [33] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees in graphs," *Acta Informatica*, vol. 15, pp. 141–145, 1981.
- [34] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
- [35] N. Singhal, L. Sahasrabuddhe, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical wdm mesh networks," *Journal of Lightwave Technology*, vol. 21, no. 11, pp. 2587–2594, Nov. 2003.
- [36] N. K. Singhal, C. Ou, and B. Mukherjee, "Cross-sharing vs. self-sharing trees for protecting multicast sessions in mesh networks," *Comput. Netw.*, vol. 50, no. 2, pp. 200–206, 2006.
- [37] M. Ali, Transmission-Efficient Design and Management of Wavelength-Routed Optical Networks. Norwell, Massachusetts: Kluwer Academic Publishers, 2001.
- [38] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, pp. 1617–1622, December 1988.

- [39] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2467–2485, 2006.
- [40] Z. Li and B. Li, "Network coding in undirected networks," in Proceedings of the 38th Annual Conference on Information Sciences, 2004.
- [41] R. Bhandari, Survivable Networks: Algorithms for Diverse Routing. Norwell, MA, USA: Kluwer Academic Publishers, 1998.
- [42] T. Rahman and G. Ellinas, "Protection of multicast sessions in wdm mesh optical networks," in *Optical Fiber Communication Conference*. (OFC/NFOEC), vol. 2, March 2005.
- [43] J. Herrera, O. Raz, E. Tangdiongga, Y. Liu, H. Mulvad, F. Ramos, J. Marti, G. Maxwell, A. Poustie, M. Hill, H. de Waardt, G. Khoe, A. Koonen, and H. Dorren, "160-gb/s all-optical packet switching over a 110-km field installed optical fiber link," *Lightwave Technology, Journal* of, vol. 26, no. 1, pp. 176–182, Jan.1, 2008.
- [44] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.
- [45] S. Bharathwaj and K. Narasimhan, "An alternate approach to modular multiplication for finite fields GF(2^m) using itoh tsujii algorithm," *The 3rd International IEEE-NEWCAS Conference*, pp. 103–105, June 2005.

- [46] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases," *Inf. Comput.*, vol. 78, no. 3, pp. 171–177, 1988.
- [47] A. Tychopoulos and O. Koufopavlou, "Optimization of the "focus" inband-fec architecture for 10-gbps sdh/sonet optical communication channels," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, April 2007, pp. 1–6.
 [48] R. Takahashi and H. Suzuki, "1-tb/s 16-b all-optical serial-to-parallel
- [48] R. Takahashi and H. Suzuki, "1-tb/s 16-b all-optical serial-to-parallel conversion using a surface-reflection optical switch," *IEEE Photonics Technology Letters*, vol. 15, no. 2, pp. 287–289, Feb. 2003.
- [49] R. Takahashi, T. Nakahara, H. Takenouchi, and H. Suzuki, "40-gbit/s label recognition and 1 × 4 self-routing using self-serial-to-parallel conversion," *IEEE Photonics Technology Letters*, vol. 16, no. 2, pp. 692– 694, Feb. 2004.
- [50] P. C. Sun, Y. T. Mazurenko, W. S. C. Chang, P. K. L. Yu, and Y. Fainman, "All-optical parallel-to-serial conversion by holographic spatial-to-temporal frequency encoding," *Optics Letters*, vol. 20, no. 16, pp. 1728–1730, August 1995.
- [51] M. Kriesell, "Edge-disjoint trees containing some given vertices in a graph," J. Comb. Theory Ser. B, vol. 88, no. 1, pp. 53–65, 2003.