Max-Flow Min-Cost Routing in a Future-Internet with Improved QoS Guarantees

Ted H. Szymanski

***Abstract*—A *Constrained Multicommodity Maximum-Flow-Minimum-Cost* routing algorithm is presented. The algorithm computes maximum-flow routings for all smooth unicast traffic demands within the *Capacity Region* of a network subject to routing cost constraints. The edge cost can be a distance, reliability, congestion or an energy metric. It is shown that every network has a finite *Bandwidth-Cost* capacity. The *Bandwidth-Distance* and the *Bandwidth-Energy* capacities are explored. The routing algorithm requires the formulation of two *Linear Programs* (LPs). The first LP finds a multicommodity Maximum-Flow, when the flows are constrained to a sub-graph of the network to enforce cost constraints. The second LP minimizes the routing cost, given that the maximum-flow is fixed. A related *Constrained Multicast-Max-Flow-Min-Cost* algorithm is also presented, to maximize the throughput of a multicast tree using network coding, subject to routing cost constraints. These algorithms have polynomial-time solutions, whereas traditional multipath routing algorithms can be NP-Hard. The addition of routing cost constraints can significantly reduce the size of the LPs, resulting in faster solutions, with lower edge utilizations and with higher energy efficiencies. The application of these algorithms to route aggregated video streams from cloud data centers in a *Future-Internet* network, with improved throughput, energy-efficiency and QoS guarantees is presented.**

***Index Terms*—Routing, multicommodity, maximum flow, minimum cost, minimum energy, Future Internet, cloud data centers, Quality of Service, QoS.**

I. Introduction

THE *Best-Effort (BE)* Internet network has evolved into universal platform for delivering new services. However, the existing BE-Internet network has several structural inefficiencies, and relies upon significant over-provisioning of bandwidth to reduce congestion and achieve weak statistical QoS guarantees [1-5]. The Internet's inefficiencies are estimated to cost hundreds of millions of dollars per year in excessive energy costs world-wide and to contribute noticeably to greenhouse gas emissions and global warming, i.e., see [6,7]. To address these problems, governments worldwide are exploring designs for the *Future Internet Network*, and are open to both *evolutionary* and *revolutionary* changes to the Internet architecture.

Cloud-based systems offer a new paradigm for service delivery [8]. Service providers can create cloud-based services by leasing cloud-based storage and computing facilities from infrastructure providers. Service providers can then use proprietary QoS routing algorithms to support their time-critical services, as is commonly done with today's VOIP providers (i.e., see www.arbinet.com). Unfortunately, the BE-Internet faces challenges a service-oriented infrastructure, due to its structural energy inefficiencies and poor QoS guarantees for time-critical services.

Over the last decade, numerous technologies have been proposed to *mitigate the poor performance of the Best-Effort Internet*, including *Peer-to-Peer* (P2P) technologies, *Network Coding*, and *Source Coding*. The goals of all these technologies include improved network throughput, energy-efficiency and QoS guarantees. *Network Coding* can potentially improve network performance by relieving the edge capacity constraints. If multiple multicast traffic flows arrive at an edge and its capacity constraint is exceeded, it is possible to forward one coded flow which obeys the edge capacity constraint. With appropriate codes, each receiver can decode and receive the multicast flow(s) [9]. Unfortunately, these technologies do not solve the fundamental structural energy inefficiencies in the underlying BE-Internet architecture.

Recently, the *Greentouch Consortium* was formed by the telecommunications industry, with a goal of achieving a 1000 times reduction in energy per bit over the Internet (www.greentouch.org). Efficient routing algorithms are fundamental to improved resource-utilization and energy-efficiency of the Internet. The goal of this paper is to explore improved routing algorithms which support Cloud-based web-services in a *Future Internet* network, to maximize the throughput of the network while simultaneously minimizing energy costs.

An Internet backbone network can be represented as a directed graph $G(V, E)$, where V is the set of routers and E is the set of fiber-optic edges. Let $|V| = N$. In current backbone networks, edge capacities are typically 10, 40 or 100 Gbps. Multiple edges often exist between nodes to provide increased capacity and reliability. The goal of a routing algorithm Γ is to find routings for traffic flows specified in an $N \times N$ *Requested Traffic-Rate-Matrix* R_R which maximize the aggregate throughput while minimizing the costs. The edge cost can be any linear function of the traffic flows on the edge, i.e., a delay, reliability, congestion, or energy metric. It is shown that every network has a finite *Bandwidth-Cost* (BC) capacity, which cannot be exceeded. Given a traffic demand matrix, a good routing algorithm will minimize the BC capacity consumed, thereby maximizing the BC capacity available for future traffic demands. Two such capacities are explored in this paper, the *Bandwidth-Distance* (BD) and the *Bandwidth-Energy* (BE) capacities.

Traditional *single-path* or *multi-path* routing algorithms will route each commodity over a single or multiple paths through the network respectively, where the path(s) are selected from

Manuscript received December 31, 2011; revised August 16 and November 27, 2012. The associate editor coordinating the review of this letter and approving it for publication was P. Popovski.

T. H. Szymanski is with the Dept. of ECE, McMaster University, Canada (e-mail: teds@mcmaster.ca).

Digital Object Identifier 10.1109/TCOMM.2013.020713.110882

a set of pre-computed candidate paths. It is well known that optimal single or multi-path routing can be NP-Hard, due to the combinatorial complexity of enumerating and processing all possible candidate paths (see section 2).

To date no polynomial-time algorithms to compute a *Multicommodity Maximum-Flow* (MMF) for multiple unicast commodities which simultaneously achieve *Minimum-Cost*, subject to cost constraints, have been proposed for the Internet (see section 2). Currently, sub-optimal routing algorithms such as the *Open Shortest Path First* (OSPF) algorithm are used to deliver services over the BE-Internet, which directly affects resource-utilization, energy-efficiency, and operating costs.

In this paper we present a *Constrained Multicommodity Max-Flow-Min-Cost* algorithm to find routings for multiple unicast commodities which maximize the aggregate flow in a graph, while minimizing the routing-cost. The edge cost can be any metric. To explore the *Bandwidth-Distance* (BD) capacity of a network, let the edge cost equal its distance. This routing-cost will tend to route commodities over shorter-distance paths, to minimize the BD-utilization. To explore the *Bandwidth-Energy* (BE) capacity of a network, let the edge cost equal the amount of energy required to transmit each Gigabit. This routing-cost will tend to route commodities over lower-energy paths, to minimize the BE-utilization.

The proposed algorithms will find routings which achieve the maximum aggregate throughput while simultaneously achieving the minimum energy cost, subject to energy cost constraints specified by the network-administrator. By relaxing the cost constraints, the *unconstrained Max-Flow-Min-Cost* algorithm will find all admissible-traffic-rate matrices within the *Capacity Region* of a network (defined in section 3). No other routing algorithms can achieve a larger aggregate multicommodity flow, or achieve a lower energy cost given this aggregate flow.

The *Constrained Multicommodity Max-Flow-Min-Cost* algorithm accepts as inputs constraints on the maximum-allowable cost of any unicast commodity flow. For every commodity to be delivered, a subgraph containing a set of candidate edges is specified. The removal of undesirable edges results in the specification of a sub-graph $G^c \in G$ for each commodity $c \in C$. The *Constrained Max-Flow-Min-Cost* algorithm consists of 2 LPs. A first LP called the *Constrained-Maximum-Flow* LP (CMF-LP) will maximize the aggregate traffic flow, subject to the constraint that every commodity is routed over its subgraph. The allowable cost for every commodity flow is constrained by selecting the subgraph $G^c(V^c, E^c)$ appropriately. An efficient algorithm to find useful subgraphs is presented. The second LP called the *Constrained-Minimum-Cost* LP (CMC-LP) will minimize the cost of the maximum aggregate traffic flow, subject to the constraint that every commodity is routed over its subgraph. There are 2 reasons to add cost constraints on each commodity; (1) Traditional MMF LPs can rapidly become intractable, even when they have polynomial-time solutions (see section 2). (2) A service provider can constrain the allowable routing cost of a commodity, to match the price its is charging to supply the service in its *Service Level Agreements* (SLAs).

Finally, we explore the use of these routing algorithms to support Cloud-based services in a recently-proposed *Future Internet* network. The search for technologies to achieve improved energy-efficiency and QoS guarantees in the Internet have existed for many years [1-7]. Most prior papers on routing assume that smooth i.e., *Constant-Bit-Rate*, traffic flows are routed. However, Internet traffic tends to be *self-similar* and highly bursty. In this paper, we summarize how self-similar video streams can be first aggregated at a Cloud data center, and then smoothened using a *token-bucket traffic shaper*, to yield a *Nearly-Constant-Bit-Rate* smoothened aggregated traffic flow. The proposed routing algorithms can then be used to route the smoothened aggregated flows over the proposed *Future Internet* network, to achieve improved throughput, energy-efficiency and QoS guarantees. We argue that these improvements are *complimentary*, i.e., that improved resource-utilization, energy-efficiency and QoS can be achieved with negligible economic cost to new routers.

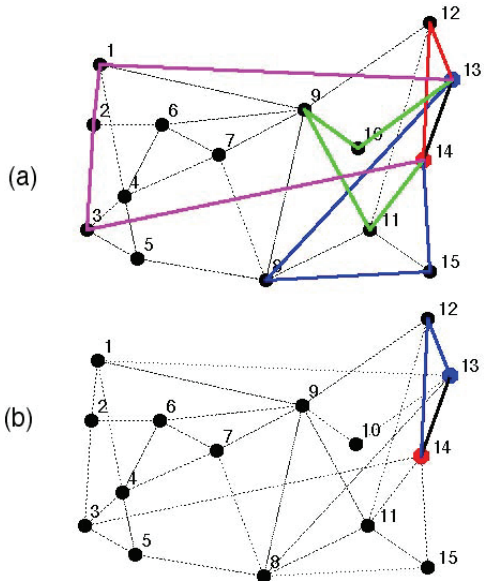

Fig. 1. Maximum-Flow routings between (13,14) in the SPRINT USA backbone network with 22 nodes and 77 edges: (a) without distance constraints; (b) with distance constraints.

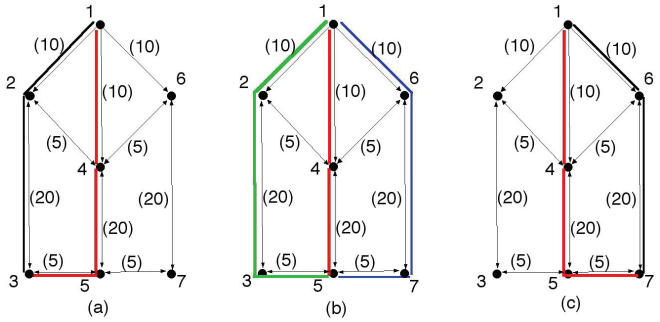

Fig. 2. 3 Maximum-flows between (1,3), (1,5) and (1,7).

The *Constrained Multicommodity Max-Flow-Min-Cost* routing algorithm applied to wireless mesh networks was presented in [18]. This paper extends [18] significantly, by (i) developing the concept of the *Bandwidth-Cost* capacity of a network and exploring two capacities, the BD and BE capacities, by (ii) developing *Multicast Max-Flow-Min-Cost* LPs in networks using network coding, by (iii) comparing

the proposed algorithm with state-of-the-art existing Internet routing algorithms, and by (iv) presenting significant experimental results on the delivery of aggregated video from Cloud-based web-servers over the *Future Internet* design proposed in [27]. The paper is organized as follows. Section 2 presents a review of routing algorithms. Section 3 formalizes the concept of Minimum-Cost routing. Section 4 presents the *Constrained Multicommodity Max-Flow-Min-Cost* algorithm. Section 5 presents routing results. Section 6 explores the routing of aggregated video streams over the proposed *Future Internet* network. Section 7 concludes the paper.

II. Review of Maximum-Flow Routings

Fig. 1 illustrates multiple paths between a (source,destination) or (s,d) pair in a network. Fig. 1a shows a SPRINT backbone network over the USA, with 15 nodes and 64 directed edges. Consider a commodity flow between nodes (13,14). There is one minimum-distance path denoted $p^* = (13, 14)$, with 1 edge and 2 routers. However, there are a combinatorially-large number of paths between these routers which vary in distance.

Many *Multicommodity Maximum Flow* (MMF) algorithms have been presented [10-13]. Let $G(V, E)$ denote a directed graph, where each edge $e = (i, j)$ has a capacity $Z(e)$. Let c denote one commodity in a set of commodities C. Each unicast commodity c has source and destination nodes denoted s_c and d_c, and a requested traffic rate W^c. Define an *Admissible Routing* as an assignment of commodity flows to the edges in G, such that no capacity constraints are violated, and flow-conservation applies at every node. The *Unsplittable Multicommodity Maximum Flow* (UMMF) problem can be stated as follows [12,13]: Does there exist an admissible routing for the commodities, where each commodity receives its requested rate W^c, and where each commodity flows over one end-to-end path? This problem is NP-Complete, since it can be restricted to include a known NP-Complete problem. When the edge capacities and requested rates are unity, the UMMF problem is equivalent to the *K-Edge-Disjoint-Paths* (K-EDP) problem, a well-known NP-complete problem [12,13]. The *K-splittable Multicommodity Maximum Flow* (KMMF) problem can be defined similarly, where each commodity is constrained to flow over at most K end-to-end paths [12,13]. This problem is also NP-Complete, as it can be restricted to include the UMMF problem. The UMMF and KMMF problems can be formulated as *Integer Programming* (IP) problems, with a combinatorially large number of possible paths to consider for each commodity.

Many *Polynomial Time Approximate Solutions* (PTAS) have been presented [11,12,13]. An algorithm yields a $(1 + \alpha)$-approximate solution to the UMMF or KMMF problem if it yields a flow rate R which is within a factor of $(1 + \alpha)$ of the maximum-flow rate. Many PTAS solutions can be found by relaxing the constraints of an IP problem to yield a *Linear-Programming* (LP) problem, which can be solved in polynomial-time.

A related but simpler path-based *Multicommodity Maximum-Flow Linear Program* (PMMF-LP) problem was presented in [10]. Given a graph $G(V, E)$ and a set of commodities C, find an admissible routing which maximizes the aggregate flow. Each commodity c may receive a fraction $f \in \{0, 1\}$ of its requested rate W^c but not more than W^c, otherwise the LP may determine a MMF where a small number of commodities receive excessively large flows. This LP allows a commodity flow to be split arbitrarily finely, and to flow over arbitrarily many paths. This LP can be solved in polynomial time [10]. Unfortunately, the number of paths to consider for each commodity grows combinatorially, leading to excessively large LPs. An iterative method to solve this problem is called *Path List Augmentation* [19], where new candidate paths are added during each iteration, and where the iterations proceed until the improvements are negligible. These iterative methods add considerable complexity to the routing algorithm, and often have difficulty converging. We illustrate another fundamental problem with the existing MMF-LPs based on [10]; a MMF solution does not necessarily minimize cost, and can yield very high edge loads unless explicit steps are taken to minimum cost.

We now summarize polynomial-time LPs related to our work, which are based on the edge-based *Multicommodity Maximum Flow Linear Program* (EMMF-LP) presented in [10]. A LP for routing multiple unicast commodities to minimize edge loads in a network was presented in [22]. This LP does not not minimize cost. A rate-optimal LP for finding a *Maximum-Flow* routing of a multicast set in networks using *Network Coding* was presented in [23]. However, this LP does not minimize cost. A cost-optimal LP for finding a *Minimum-Cost* routing of a multicast set in a network using *Network Coding* was presented in [24], under the assumption of 'linear separable commodity costs'. However, this LP does not find a maximum-flow rate, and it does not achieve the true minimal cost under all conditions. It is shown in this paper that lower costs can be achieved. A LP for maximizing throughput subject to QoS constraints in a network using *Network Coding* was presented in [25]. However, this LP does not minimize cost. All these prior LPs can become intractable, due to the large number of variables to be considered. Consider a network $G(V, E)$ with $|V| = N$ and $|E| = M$. There are potentially $N \times (N - 1)$ commodities to be routed. The traditional EMMF-LPs require a LP variable for every commodity flow on every edge, i.e., potentially $N \times (N-1) \times M$ variables. For a network with 26 nodes and 84 edges, the EMMF-LP can require 54,600 variables, which can be intractable for real-time Internet routing.

III. Single Path and Multipath Routing

The existing OSPF Internet routing algorithm finds a shortest weight path between an (s,d) pair in a weighted graph G(V,E). The network administrator can assign the 'cost' or 'weight' associated with each edge. The OSPF edge cost can be defined as the inverse available bandwidth on the edge, so the edge cost reflects the idealized MM1 queueing delay (assuming all traffic is Poisson-distributed), and the shortest weight path is the shortest delay path. In practice, OSPF updates the routing tables in a BE-Internet router periodically, to identify the current shortest delay path for each remote router. The OSPF paths can change frequently, due to transient network congestion associated with bursty BE traffic flows.

OSPF supports an *Equal Cost Multi-Path* routing (EQMP) option but this not frequently used, as the OSPF routing paths change frequently and the delay on any OSPF path can be highly variable. The OSPF algorithm is frequently used within single *Autonomous Systems* in the BE-Internet [21].

Routing algorithms based on shortest paths are often used in VLSI integrated circuit design [16,17]. For example, the BoxRouter CAD tool can improve throughput while decreasing wire-lengths in VLSI circuits using shortest path algorithms and integer linear programs [16].

A path-oriented *K-path Multicommodity Maximum Flow Integer Program* (KP-MMF-IP) is shown in Eq. (1). Let $c \in C$ denote one commodity in a set C. Each commodity c has a requested traffic rate W^c which can flow over up to K end-to-end paths between the pair (s_c,d_c). Let P^c be a set of candidate paths associated with a commodity c, with cardinality $|P^c| \geq K$, and let $p \in P^c$ denote one path in P^c. Each commodity $c \in C$ has a vector of binary decision variables B^c, where $B^c(p)$ denotes the binary variable associated with path $p \in P^c$. The binary decision variable $B^c(p)$ is asserted if any positive fraction of the commodity c is routed over the corresponding end-to-end path $p \in P^c$. Let $r^c(e)$ and $r^c(p)$ denote the rate of flow of commodity c over edge e and path p respectively. Let $Z(e)$ denote the capacity of edge e.

$$\text{maximize } r^* \tag{1}$$

$$\begin{array}{ccccc} r^c(p) & \geq 0 & \forall c \in C, \forall p \in P^c & (1.1) \\ r^c(e) & \leq Z(e) & \forall c \in C, \forall e \in E & (1.2) \\ \sum_{c \in C} r^c(e) & \leq Z(e) & \forall e \in E & (1.3) \\ B^c(p) & \in \{0,1\} & \forall c \in C, \forall p \in P^c & (1.4) \\ \sum_{p \in P^c} B^c(p) \cdot r^c(p) & = W^c & \forall c \in C & (1.5) \\ \sum_{p \in P^c} B^c(p) & \leq K & \forall c \in C, \forall p \in P^c & (1.6) \end{array}$$

$$r^* = \sum_{c \in C} \sum_{p \in P^c} B^c(p) \cdot r^c(p)$$

Constraints 1.2-1.3 enforce edge capacity constraints. Constraint 1.5 asserts that the requested flow rate W^c for commodity c is met. Constraint 1.6 asserts that one commodity flows over at most K end-to-end paths. To find a true MMF in $G(V, E)$, all possible end-to-end paths must be considered in the set P^c for each $c \in C$, and the problem is in NP.

Consider a related but simpler *Linear Program* to find a MMF given a fixed set of K candidate end-to-end paths P^c for each commodity $c \in C$. Call this problem the *K-Shortest-Path LP* (KSP-LP), which can be stated as follows:

$$\text{maximize } r^* \tag{2}$$

$$\begin{array}{ccccc} r^c(p) & \geq 0 & \forall c \in C, \forall p \in P^c & (2.1) \\ r^c(e) & \leq Z(e) & \forall c \in C, \forall eE & (2.2) \\ \sum_{c \in C} r^c(e) & \leq Z(e) & \forall eE & (2.3) \\ \sum_{p \in P^c} r^c(p) & \leq W^c & \forall c \in C & (2.4) \end{array}$$

$$r^* = \sum_{c \in C} \sum_{p \in P^c} r^c(p)$$

This LP can be solved in polynomial time, and will yield a MMF considering only the fixed set of K shortest paths for each commodity $c \in C$. The solution of the KSP-LP in Eq. 2 only yields an approximation solution to the K-MMF-IP in Eq. 1, since the LP only considers a small subset of paths P^c for each commodity c. Furthermore, Eq. (2) limits the rate allocated to any commodity c to $\leq W^c$, otherwise the LP may find a MMF where a small number of commodities receive excessively large flows, and where all other commodities remain unserviced. Nevertheless, the KSP-LP in Eq. 2 can yield good solutions to the routing problem in the Internet, and is considered in section VI.

IV. Defining the Bandwidth-Cost Capacity

Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{Z}$ denote the set of integers. Define an integer matrix $E \in \mathbb{Z}^{N \times N}$, where $E(i,j) = 1$ if vertices i and j are joined by an edge, otherwise 0. Define a non-negative matrix $Z \in \mathbb{R}^{N \times N}$, where $Z(i,j)$ represents the capacity constraint of edge (i,j), i.e., typically 10, 40 or 100 Gbps. (Let $Z(i,i) = 0$). Define a cost-matrix $\Theta \in \mathbb{R}^{N \times N}$, where $\Theta(i,j)$ represents the cost of edge (i,j). (To avoid too many symbols, let E and Θ denote a set or a matrix, which is clear from the context.) A path $P = (v_1, ..., v_n)$ is a set of vertices representing adjacent routers. Let the cost of a path be the sum of the edge costs in the path.

In a network $G(V, E)$ with N nodes, a set of $N \times (N-1)$ requested commodity flow rates can be specified between all pairs of distinct nodes. Define a *Requested-Rate-Matrix* $R_R \in \mathbb{R}^{(N \times N)}$, where $R_R(i,j)$ denotes the requested rate for a smooth (i.e., constant-bit-rate) commodity flow between nodes (i,j), where $R_R(i,i) = 0$. Define an *Admissible-Rate-Matrix* $R_A \in \mathbb{R}^{(N \times N)}$, where $R_A(i,j)$ denotes an admissible commodity flow rate between nodes (i,j) determined by some routing algorithm Γ, without violating any edge capacity constraints. A traffic rate matrix is *admissible* if its requested rates can be simultaneously supported in the network, without violating any edge capacity constraints. Given a network $G(V, E)$ with edge capacity constraint matrix Z and edge cost matrix Θ, each admissible *Traffic-Rate-Matrix* R_A will define one point in $N \times (N-1)$-dimensional space. The set of all possible admissible matrices R_A will define a polytope in $N \times (N-1)$-dimensional space. The convex hull of the polytope defines the *Capacity Region* of the network. This region has also been called the *Stability Region* or the *Throughput Region* in various papers [14,15].

Any network $G(V, E)$ can be viewed as having a finite amount of resources, expressed as a *Bandwidth-Cost (BC) Capacity*. The cost of an edge can be any metric. The *BC-capacity* of a network can be defined as

$$BC = \sum_{i=1}^{N} \sum_{j=1}^{N} Z(i,j) \cdot \Theta(i,j) \tag{3}$$

Given a traffic demand matrix, an optimal routing should minimize the BC-capacity consumed by the routed traffic, thereby leaving more unused capacity for future traffic demands. Two BC capacities will be explored, the *Bandwidth-Distance* (BD) and the *Bandwidth-Energy* (BE) capacities. The units of the *BD-capacity* are *Gigabit-miles per second (Gbmps)*. Referring to the SPRINT topology in Fig. 1a, let every directed link have a capacity of 1 Gbps. The *BD-capacity* of the SPRINT network is 465.5K Gbmps. Similarly, the units of the *BE-capacity* are *Gigabit-Joules per second (Gbjps)*.

Define the BC-utilization consumed by a commodity c flowing over a path p with rate $r^c(p)$ as

$$BC(p, r^c(p)) = \sum_{e \in p} r^c(p) \cdot \Theta(s(e), d(e)) \quad (4)$$

where $s(e)$ and $d(e)$ represent the source and destination vertices of edge e. The minimum BC utilization consumed per unit flow of commodity c occurs over its shortest path p^*, and is denoted $BC(p^*, 1)$. Consider a commodity c routed over several paths $p \in P^c$ between pair (s_c, d_c) in $G(V, E)$. The rate assigned to each path is $r^c(p)$. Define the *BC-Expansion* of commodity c as the ratio

$$\sum_{p \in P^c} BC(p, r^c(p)) / \left(\sum_{p \in P^c} r^c(p) \cdot BC(p^*, 1) \right) \quad (5)$$

The expansion is the ratio of resources needed to meet the commodity demand in $G(V, E)$, relative to the resources needed to meet the commodity demand assuming it could be satisfied by a minimum-cost path. The *BC-expansion* illustrates the effectiveness of a given topology $G(V, E)$ to realize a particular commodity. A *BC-expansion* close to unity indicates the network is well-suited the handle the commodity, as it flows over mostly shortest-cost path(s). A *BC-expansion* much larger than unity indicates the network is poorly-suited the handle the commodity, i.e., significant amounts of the commodity flow over sub-optimal higher-cost paths.

Fig. 1 also illustrates several concepts related to the BD-utilization of a *Maximum-Flow* routing. Let the capacity of each edge in the SPRINT topology shown in Fig. 1a be $Z = 1$ Gbps. A *Maximum-Flow* F between nodes (13,14) has rate 5 Gbps, and uses 5 end-to-end paths with rates 1 Gbps each as shown in bold between nodes (13,14). The distances of these 5 end-to-end paths are (610, 1,423, 2,854, 3,830, 5,939) miles respectively. The BD-utilization over the shortest-distance path is 610 Gbmps. No other path can deliver 1 Gbps with lower BD-utilization. The BD-utilization over the longest path in F, $P = (13, 1, 2, 3, 14)$ is 5,939 Gbmps, about 9.5 times the cost of the first path to deliver 1 Gbps. The BD-utilization of the Maximum-Flow F is 14,655 Gbmps. The *BD-expansion* for this maximum flow is 4.8, i.e., this routing requires 4.8 times the resources used in the minimum-distance path to deliver each Gbps. The *Maximum-Flow* F removes considerable resources from the network. These resources are inefficiently used by F, and cannot be used by other commodity flows which may have considerably more efficient routings. It may be desirable to constrain the maximum distance that any commodity may use, to reduce inefficient resource usage.

Fig. 1b illustrates a *Maximum-Flow* routing between nodes (13,14) when the path distance is constrained to be within 2,000 miles of the minimum-distance path. Most edges in the SPRINT topology have been removed from consideration, and only the first 2 paths meet the distance constraint. This *Constrained-Maximum-Flow* delivers 2 Gbps and consumes a *BD-utilization* of 2,034 Gbmps. The *BD-expansion* is 1.66, i.e., this routing requires 66% more resources than the minimum-distance path to deliver each Gbps, considerably better resource-efficiency than the unconstrained *Maximum-Flow*.

A. The Energy Cost of Multicast Routings

Ahlswede et al [9] have shown that *Network Coding* allows a network to support a multicast routing with the maximum-achievable rate. However, they did not consider the costs of multicast trees using *Network Coding* (NC). Two papers [23,24] have established in theory that optimum multicast routings with the maximum-achievable rate and minimum-achievable cost can be found in polynomial-time when NC is used. In this section, we show that the polynomial-time algorithms in [23,24] do not guarantee multicast routings with minimal cost under all conditions. We show that multicast routings with the same rate can have significantly different *Bandwidth-Energy* costs.

Fig. 2 shows a 7 node network. Consider an Internet backbone network, and let the capacity of each edge be Z=1 Gbps. Let each edge have an energy-cost in joules per Gbps as shown in Fig. 2. A multicast set M from node $m(0)$ to nodes $m(1), m(2), ..., m(L)$ is represented by a vector $M \in \mathbb{Z}^{1 \times (L+1)}$. Consider multicast set $M = [1, 3, 5, 7]$ from node 1 to nodes $(3, 5, 7)$.

Let $H(s, d)$ denote the single-commodity maximum flow rate in a graph $G(V, E)$ between the node pair (s, d). Referring to Fig. 2, the three *Max-Flow* rates $(H(1,3), H(1,5), H(1,7)) = (2, 3, 2)$ Gbps respectively. According to [9], a multicast set M can support rate $R \leq R^*$, where the maximum-achievable multicast rate is $R^* = min(H(m(0), m(1)) ..., H(m(0), m(L))$. Therefore, the graph in Fig. 2 can support a multicast set $M = [1, 3, 5, 7]$ with R^*= 2 Gbps, and NC can be used to achieve this rate.

There are 3 important cases in any multicast set; (i) $R \leq Z$, (ii) $Z < R < R^*$, and (iii) $R = R^*$. In an Internet backbone network where the link capacities are typically 10...100 Gbps, the relevant case is usually case (i). When $R \leq Z$, the minimum-cost routing of a multicast set is given by the Steiner tree, and NC cannot improve upon this cost. Unfortunately, finding the Steiner tree is an NP-hard problem. When $Z < R \leq R^*$, finding a minimum-cost routing of multicast set (when NC is not used) entails solving the *Steiner tree packing* problem, which packs multiple Steiner trees into G to realize rate R, and is also an NP-hard problem.

A rate-optimal polynomial-time LP to find a *Max-Flow* routing of a multicast set using NC was presented in [23]. Each set M is represented as multiple unicast flow problems in one large LP which maximizes the aggregate flow. However, while the resulting routings do achieve the optimum Maximum-Flow, they do not necessarily achieve Minimum-Cost. Using the LP in [23] on Fig. 2, the *BE*-cost consumed by the *Max-Flow* multicast routing with R=2 Gbps is 132.4 Gbjps. (NC is not necessary but can be used in this example.) The BE-cost is sub-optimal due to the existence of traffic cycles. To minimize the BE-cost, a cost minimization step is necessary.

A cost-optimal polynomial-time LP to find a *Min-Cost* routing of a multicast set using NC was presented in [24], Eq. 3. Each set M is represented as multiple unicast flow problems in one large LP which minimizes the cost given

the specified multicast rate R. However, we observe that the cost of the multicast routings is minimized only when the rate $R = R^*$, in which case the assumption of *'separable costs'* made in [24] is valid. Otherwise, lower cost multicast routings which do not use NC exist. For example, with $R = Z$, the LP in [24] yields a multicast routing with a cost of 90 Gbjps (3 separable minimum-cost paths costing 30 Gbjps each). NC does not improve this solution, as no flows can be coded. However, the *Steiner tree* from node 1 through node 5 to destinations (3,5,7), supports rate R=1 Gbps, and consumes a *BE-product* of 40 Gbjps, less than half the LP cost. For case 2 with R=1.6 Gbps, the LP in [24] yields a cost of 99 Gbjps after NC. However, an optimal routing that uses the Steiner tree yields a cost of 79 Gbjps. For $R = R^*$= 2 Gbps, the LP in [24] computes a multicast routing with a BE-cost = 105 Gbjps after NC. Only for case 3 is the BE-cost optimal.

In summary, we have shown that the multicast routings generated by the polynomial-time LPs in [23,24] followed by the use of NC do not necessarily have minimum cost, measured as a *Bandwidth-Cost* product or *Bandwidth-Energy* product. Nevertheless, the polynomial-time LPs in [23,24] together can provide fast approximate solutions to the true *Multicast Max-Flow-Min-Cost* problem, which exploit the Steiner tree when $R < R^*$.

Implications: In [36,37], polynomial-time approximation algorithms were used to compute sub-optimal routings for multicast sets in a VLSI circuit when $R \leq Z$, and NC was then applied to potentially reduce the costs. (The routing algorithms used linear programs, heuristics and artificial intelligence.) The routing algorithms applied to the case $R \leq Z < R^*$, in which case the Steiner tree is the optimal minimum-cost solution. There was no discussion of whether the routings achieved minimal cost, and the results were not compared to routings that used the Steiner trees, so that the degree of sub-optimality was not quantified. VLSI ICs such as the Intel iCore processors are usually mass-produced to yield tens of thousands of ICs, to offset the considerable VLSI design costs. Furthermore, these ICs are often used in products for several years. The use of a sub-optimal multicast tree with a sub-optimal BE-cost can result in high recurring energy costs, spread over tens of thousands of ICs over several years. Given the high energy costs associated with sub-optimal solutions, when $R < R^*$ it may be advantageous to employ (a) more complex polynomial-time algorithms, and (b) combinatorial algorithms such as branch-and-bound, to perform a more thorough (but non-exhaustive) search the solution-space, in the search for energy-efficient multicast routings.

V. The Constrained Max-Flow-Min-Cost Algorithm

This section presents the *Constrained Max-Flow-Min-Cost* algorithm. Each unicast commodity is constrained to flow over a set of feasible edges. An efficient algorithm to determine a feasible edge set for each commodity is shown in Fig. 3. Given a requested traffic rate matrix $R_R \in \mathbb{R}^{N \times N}$, there $\leq N \times (N-1)$ unicast commodities to be routed. The objective is to compute a subgraph $G^c(V^c, E^c) \in G(V, E)$ for each commodity $c \in C$, with distance constraints.

Referring to Fig. 3, for every node in G the algorithm initially computes a minimum-cost path to every other node using Dijkstra's algorithm. Dijkstra's algorithm will yield a tree, with minimum-cost paths from the root to the leaves. The complexity of Dijkstra's algorithm is $O(|E|+|V|log|V|)$. For every commodity c to be routed between a pair (s,d), the algorithm initializes a set of candidate nodes and a set of candidate edges to be NULL. The algorithm then visits all intermediate nodes $v \in V$. Let $M(s,v)$ denote the length of a minimum-cost path between (s,v). If $M(s,v)$ + $M(v,d) \leq M(s,d)$ plus a *Cost-Threshold* T, then the node v is included in a set of feasible vertices V^c for the commodity. The set of feasible edges E^c consists of the edges in the subgraph V^c, i.e., the set of edges in E whose endpoints are in V^c which meet the cost constraints. The computation of the subgraph for each commodity has complexity $O(|V|)$, assuming matrix M is precomputed. Once a subgraph G^c is computed for every $c \in C$, the *Constrained Minimum-Cost-Maximum-Flow* algorithm, consisting of two LPs, can be formulated.

A. The Constrained-Maximum-Flow LP - LP-#1

The *Constrained-Maximum-Flow* (CMF) LP for commodities $c \in C$ is given by Eq. 6. Each commodity c has a source-destination pair (s_c, d_c) and a requested traffic demand W^c. The LP will maximize the aggregate flow over all commodities, while attempting to provide each commodity with its requested rate (but not more). Due to capacity constraints some commodities may only receive a fraction of their requested rates.

$$\text{Maximize: } r^* \tag{6}$$

Subject to:

$$\begin{array}{lll}
0 \leq r^c(e) & \forall c \in C, \forall e \in E^c & (6.1)\\
r^c(e) \leq Z(e) & \forall c \in C, \forall e \in E^c & (6.2)\\
\sum_{c \in C} r^c(e) \leq Z(e) & \forall c \in C, \forall e \in E & (6.3)\\
r^c_{in}(v) = r^c_{out}(v) & \forall c \in C, \forall v \in V^c - (s_c, d_c) & (6.4)\\
r^c_{in}(s_c) = 0 & \forall c \in C & (6.5)\\
r^c_{out}(d_c) = 0 & \forall c \in C & (6.6)\\
r^c_{out}(s_c) \leq W^c & \forall c \in C & (6.7)\\
r^c_{out}(s_c) \geq L^c & \forall c \in C & (6.8)
\end{array}$$

$$r^* = \sum_{c \in C} r^c_{in}(d_c)$$

Let $r^c(e)$ denote the flow rate of commodity c on edge e. Let $r^c_{in}(v)$ and $r^c_{out}(v)$ denote the total flow rate into / out of node v due to commodity c, respectively. Constraint 6.3 requires that the sum of all commodity flow-rates over an edge e is $\leq$ the edge capacity $Z(e)$. Flow-conservation constraints 6.4-6.6 are restricted to the subgraph G^c for each commodity c. Constraints 6.5 and 6.6 for all $c \in C$ can also be merged into one constraint, to reduce the problem size substantially. Constraints 6.7 and 6.8 ensure that each commodity receives a rate at least L^c and at most W^c. The LP is solved and the maximum-flows are determined.

```
for s ∈ V
      M(i,:) = minimum distance vector
               (using Dijkstra's algorithm)
end;
for c ∈ C
  Vc= NULL;
  for v ∈ G
       if (M(sc,v) + M(v,dc)) ≤ (M(sc, dc) + T)
          V'= V' ⋃ v
       end;
   end;
end;
Ec = NULL;
for u ∈ Vc, v ∈ Vc
   if ((u,v) ∈ E) & (M(sc,u)+M(v,dc))≤ (M(sc, dc)+T)
       Ec = Ec ⋃ (u,v)
    end;
end;
```

Fig. 3. The 'Sub-graph' flow-chart to find subgraphs for commodity flows.

B. The Constrained-Minimum-Cost LP - LP-#2

To obtain the minimum-achievable routing-cost, a second cost-minimization LP is formulated. Let Λ^c be the maximum-flow rate of commodity $c \in C$ between (s^c,d^c), determined by the *Constrained-Maximum-Flow* LP. Let the linear cost associated with every edge $e \in E$ be given by $\Theta(e)$. (The cost can be any metric, i.e., a distance, reliability, congestion or energy metric). The following *Constrained-Minimum-Cost* (CMC) LP given in Eq. 7 will minimize the cost of the *Maximum-Flow* determined from the CMF LP:

$$\text{Minimize: } y^* \tag{7}$$

Subject to:

$$\begin{array}{lll} r^c_{out}(s_c) & = \Lambda^c +/- \epsilon & \forall c \in C, \forall e \in E^c \quad (7.1) \\ \sum_{c\in C} r^c(e) & \leq Z(e) & \forall c \in C, \forall e \in E \quad (7.2) \\ r^c_{in}(v) & = r^c_{out}(v) & \forall c \in C, \forall v \in V' \quad (7.3) \\ r^c_{in}(s_c) & = 0 & \forall c \in C \quad (7.4) \\ r^c_{out}(d_c) & = 0 & \forall c \in C \quad (7.5) \end{array}$$

$$y^* = \sum_{c\in C} \sum_{e \in E} r^c_e \times \Theta(e)$$

where $V' = V - (s_c, d_c)$. Constraint 7.1 requires that the maximum-flow rate for commodity c equals the value Λ^c determined by the CMF LP, plus/minus a threshold ϵ (typically 10^{-4}). The remaining constraints are similar to those in Eq. 6. The objective function is to minimize the *Bandwidth-Cost* utilization consumed by the routing, thereby maximizing the remaining BC capacity available for future traffic demands.

The Successive Relaxation Algorithm: The LPs considered in [10,22,23,24,25] can grow to be very large and potentially intractable. The introduction of cost-constraints will limit the size of the LPs, by restricting the number of edges to consider for each commodity. Consider the following algorithm, called the *Successive Relaxation Algorithm*: Using the distance cost, initially a problem is solved with a small *distance-threshold* T (i.e., D^*/64, where D^* is the diameter of the graph.) If the demands are satisfied, then the routing is complete. Otherwise, the *distance-threshold* T can be doubled, and the LP can be solved again. The algorithm iterates until the problem is solved. The choice $T = D^*/64$ implies that the cost constraints are effectively removed in the 6-th iteration. This algorithm can yield approximate solutions for large graphs, which may otherwise be intractable.

C. The Constrained Multicast Max-Flow-Min-Cost Algorithm

In this section, the rate-optimal LP in [23] and the cost-optimal LP in [24] to perform multicast routings with Network Coding are modified to include explicit cost constraints. A multicast set M is a vector $[m(0)....m(L)]$. In both LPs, the set M is represented as L distinct unicast flows $c \in C = \{(m(0), m(1)) ,..., (m(0), m(L))\}$, as described earlier. The rate-optimal LP in [23] finds a *Maximum-Flow* for the multicast set, under the constraint that the L distinct unicast rates are maximized and equal. Network coding can be used to resolve any edge capacity constraint violations. The maximum-achievable multicast rate is denoted R^*.

We follow the same approach as [23,24], and represent a multicast set M as L distinct unicast flows. The CMF LP in Eq. 6 can be used, with constraints 6.3, 6.7 and 6.8 removed. Constraint 6.3 enforces an edge capacity constraint $Z(e)$ for edge e. This constraint can be replaced by Eq. 8, which states that the sum of all commodity flow-rates over an edge e in G is less than infinity. (This constraint is explicitly shown in Eq. 8 for pedagogical purposes.) In addition, a new constraint in Eq. 9 replaces constraints 6.7 and 6.8, to ensure that the flow to each destination in the multicast set is equal.

$$\sum_{c\in C} r^c(e) \quad \leq \infty \quad \forall c \in C, \forall e \in E^c \quad (8)$$

$$r^c_{out}(s_c) \quad = r^b_{out}(s_b) \quad \forall c, b \in C, \forall e \in E^c \quad (9)$$

The new CMF LP in Eq. 6 (with Eq. 8 and 9 replacing constraints 6.3, 6.7 and 6.8) will yield the same *Maximum-Flow* multicast rate R^* as the LP in [23], when the cost constraints are relaxed. When the cost constraints are enforced, the number of variables in the LP is reduced considerably, often by orders of magnitude for large networks, and the execution time can decrease similarly.

To find a *Minimum-Cost* routing for a multicast set with the maximum-achievable rate R^*, the *Constrained-Minimum-Cost* LP in Eq. 7 can be used, where Eq. 10 replaces constraint 7.2:

$$\sum_{c\in C} r^c(e) \quad \leq \infty \quad \forall c \in C, \forall e \in E^c \quad (10)$$

Constraint 10 removes the edge capacity constraints, since *Network Coding* can resolve any edge capacity constraint violations. The modified CMC LP in Eq. 7 (with constraint 7.2 removed) will yield the same minimum-cost routing as the LP in [24] for the case $R = R^*$, when the cost-constraints are relaxed. When cost constraints are enforced, the number of variables to be solved in the LP can be reduced considerably, often by orders of magnitude for large networks.

When network coding is used and the edge capacity constraints are relaxed, we further observe that the LPs in [23,24] and our constrained versions can be simplified. These LPs effectively find multiple independent minimum-cost paths from the source to the destinations of the multicast set, when rate $R = R^*$. Therefore, an LP is not needed as Dijkstra's algorithm can be used to solve the same problems.

In summary, the rate-optimal LP in [23] and the cost-optimal LP in [24] and the versions presented in this paper can be used to find *Multicast Max-Flow-Min-Energy* routings when the multicast rate is maximized ($R = R^*$). However, these LPs do not find the *Multicast Max-Flow-Min-Energy* routings when $R < R^*$, as explained in section IVa. Nevertheless, the multicast LPs in [23,24] and the constrained versions presented here may yield acceptable approximate solutions for multicast routings in polynomial time when $R < R^*$.

D. Properties

Some properties of the *Multicommodity Max-Flow-Min-Cost* algorithm are summarized.

Lemma 1: Let s and d be the source and sink of a unicast traffic flow in $G(V, E)$. Then there exists a *Maximum-Flow* F between (s,d) in G which can be expressed as the sum of a number of flows each consisting of a simple path from s to d (i.e., a directed path without cycles).

Proof: The proof is similar to the proof of Lemma 1 in Ahlswede et. al. [9]. Let F be a *Maximum-Flow* from s to d in $G(V, E)$ which does not contain a directed cycle. Let $P(i)$ be any positive path from s to d, and let $c(i)$ be the minimum value of F assigned to an edge in $P(i)$. Let F^i be the flow from s to d along $P(i)$ with rate $c(i)$. Subtracting F^i from F yields $F - F^i$, a flow from s to d which does not contain a directed cycle. Apply the same procedure until F is reduced to zero. The lemma is proved. ■

Property 1: By setting the cost-threshold T for every unicast commodity $c \in C$ equal to ∞, the *Constrained-Maximum-Flow* LP in Eq. 6 finds the *Maximum-Flow* F of all unicast commodities, subject to edge capacity constraints. No other routing algorithm can achieve a larger *Maximum-Flow*.

Proof: By contradiction. By setting the cost-threshold to ∞, then the subgraph associated with every commodity is the full graph $G(V, E)$, and there are no additional cost constraints associated with any commodity. The *Constrained Maximum-Flow* LP in Eq. 5 reduces to the unconstrained edge-based MMF LP in [10,23]. ■

Property 2: The solution to the *Constrained-Minimum-Cost* LP in Eq. 7, given the solution vector r of the *Constrained Maximum-Flow* LP in Eq. 5 for which the cost-threshold T for every commodity equals ∞, the cost of the *Maximum-Flow* is minimized. No other routing algorithm can achieve a lower cost, given the rate constraints specified in the *Constrained-Maximum-Flow* LP in Eq. 6.

Proof: By contradiction. If it is not true, then either the solution to unconstrained *Maximum-Flow* LP in Eq. 6 does not yield a maximum-flow, or the solution to unconstrained *Minimum-Cost* LP in Eq. 7 did not yield a minimum-cost. When the edge cost is the energy per Gbps, no other routing algorithm can achieve a lower *Bandwidth-Energy* cost. ■

Theorem 1: Given a graph $G(V, E)$ where $|V| = N$, given a non-negative edge capacity matrix $Z \in \mathbb{R}^{N\times N}$, given a non-negative edge cost matrix $\Theta \in \mathbb{R}^{N\times N}$, given a matrix of minimum-cost path values between (s,d) pairs $M \in \mathbb{R}^{N\times N}$ computed from matrix Θ, and given a requested traffic demand matrix $R_R \in \mathbb{R}^{N\times N}$, then a necessary condition for the R_R to lie within the *Capacity Region* of G is that the BC-utilization demanded by R_R must not exceed the BC-capacity of G(V,E), i.e.,

$$\sum_{s=1}^{N}\sum_{d=1}^{N} R_R(s,d)M(s,d) \le \sum_{i=1}^{N}\sum_{j=1}^{N} Z(i,j)\Theta(i,j). \quad (11)$$

Proof: The proof follows the proof of lemma 1, and is by contradiction. Suppose a matrix R_R which violates Eq. 11 can be routed. Consider routing commodities iteratively along minimum-cost paths, decrementing the remaining capacities of all traversed edges appropriately after each commodity is routed. At some point, the remaining BC-capacity on the RHS of Eq. 11 will be exhausted before all the commodities have been routed. The routing of any remaining commodity along a minimum-cost path will result in an edge capacity violation. ■

Theorem 1 provides a necessary condition on the admissibility of any requested traffic rate matrix R_R. If the *BC-utilization* demanded by R_R exceeds the *BC-capacity* of the network, then the matrix R_R is not within the *Capacity Region* of the network, and it cannot be fully routed. It also yields a simple test to estimate the resource-utilization of a requested traffic rate matrix. If the BC-utilization demanded by the matrix is less than $\approx 40\%$ of the BC-capacity, then the matrix can likely be routed along predominantly minimum-distance paths, and the distance threshold used in the CMF and CMC LPs can be relatively small.

VI. Routing Results

This section summarizes the results of the LPs for an expanded version of backbone USA topology shown in Fig. 1, with 26 nodes and 84 edges, where the capacity of every edge = 40 Gbps (see [28]). The *BD-Capacity* of the network is 1.522M Gbmps. From theorem 1, a necessary condition for a traffic pattern to be achievable is that the demanded *BD-utilization* must not exceed 1.522M Gbmps.

One hundred *Traffic-Demand-Matrices* were generated for 2 scenerios, the *light-load* and *heavy-load* scenarios. For the light-load scenario, each traffic demand matrix consists of 8 randomly generated permutations, where each (s,d) pair has a demand of 1.2 Gbps. For the heavy-load scenario, each traffic demand matrix consists of 16 randomly generated permutations, where each (s,d) pair has a demand of 1.6 Gbps. The 100 traffic demand matrices where routed on a laptop processor, with 2 processing cores with 2.8 GHz clock-rates, and 8 Gbytes of main memory.

Table 1 illustrates the results of the OSPF algorithm for the light-load scenario. The routing algorithm is shown in Column 1. The distance-threshold T for the CMF or CMC algorithms is shown in brackets. The demanded *BD-utilization* (before routing) is denoted by BD(D) in column 2. The *BD-utilization* consumed after routing is denoted by BD(U) in column 3. The processor execution time is denoted by ExT in column 4. Three edge weights are used by the OSPF algorithm to find shortest paths. In OSPF(DEL), the edge weight equals its inverse available edge bandwidth, so that the shortest-paths are the lowest-delay paths (assuming the traffic was Poisson-distributed). In OSPF(DIST), the edge weight equals the edge distance (in miles). In OSPF(HOP), each edge has a weight of

TABLE I
ROUTING RESULTS, LIGHT-LOAD TRAFFIC MODEL

Algorithm	BD(D) (Gbmps)	BD(U) (Gbmps)	ExT (sec)	Flow percent	edge load (%)
OSPF (DEL)	348K	385K	1.07	100 %	27.2 %
OSPF (DIST)	”	348K	1.03	100 %	25.9 %
OSPF (HOP)	”	399K	1.04	100 %	26.5 %
KSP (8 paths)	”	587K	0.59	100 %	40.3 %
KSP (16 paths)	”	561K	0.73	100 %	44.4 %
EMMF	348K	1,468K	4.29	100 %	96.4 %
CMF-CMC(8000)	”	348K	8.09	100 %	25.9 %
CMF-CMC (1000)	”	348K	3.95	100 %	25.9 %
CMC (100)	”	348K	0.30	100 %	25.9 %

TABLE II
ROUTING RESULTS, HEAVY-LOAD MODEL

Algorithm	BD(D) (Gbmps)	BD(U) (Gbmps)	ExT (sec)	Flow percent	edge load (%)
OSPF (DEL)	917K	1,096K	2.06	96.9 %	71.6 %
OSPF (DIST)	”	940K	2.30	92.4 %	66.0 %
OSPF (HOP)	”	1,001K	2.17	95.3 %	64.9 %
KSP (8)	”	1,055K	1.20	76.8 %	67.8 %
KSP (16)	”	1,266K	2.02	99.4 %	86.7 %
CMF-CMC (4000)	917K	933K	23.84	99.6 %	64.3 %
CMF-CMC (1000)	”	933K	8.17	99.4 %	64.3 %
CMC (1000)	”	933K	4.24	99.4 %	64.3 %

TABLE III
SIZE OF LPS, HEAVY-LOAD TRAFFIC MODEL

Algorithm	Flow	Variables	EQ Cons.	LE Cons.
EMMF	NA	26,124	7,465	395
CMF-CMC (4000)	99.6 %	23,958	6,895	392
CMC (2000)	99.5 %	18,345	5,358	389
CMC (1000)	99.4 %	11,071	3,371	389
CMC (500)	99.2 %	6,220	2,143	389
CMC (250)	90.7 %	2,908	1,342	389

1. For the light-load scenario, all 3 OSPF algorithms achieve 100% of the traffic demands. OSPF(DIST) also consumes the minimum BD-utilization. The BE-Internet is generally over-provisioned and operates at a light loads on average [1-3]. According to table 1, OSPF has excellent performance at light loads, achieving 100% of the traffic demands with low cost (BD-utilization). The computational requirement of OSPF is small and is distributed amongst the BE-Internet routers. Table 1 also illustrates the results of the *K-Shortest-Paths* LPs, for K=(8,16) paths. Both achieve 100% of the demand, but they incur higher BD-utilizations and higher average edge loads than OSPF, since the KSP-LP has no incentive to minimize cost. (The edge loads can be improved by adding a cost-minimization LP after the KSP-LP).

Table 1 also illustrates the results of 3 algorithms for the light-load scenario, (i) the edge-based EMMF-LP in [10,23], (ii) the proposed CMF-CMC-LP with distance threshold = (8000,1000) miles, and (iii) the proposed CMC-LP with distance threshold = 100 miles. All three algorithms achieve 100% of the traffic demand. The traditional EMMF-LP achieves a very poor average edge load of 96.4%, illustrating an inefficient routing with traffic cycles, since this LP has no incentive to minimize cost. The CMF-CMC-LPs achieve 100% of demand, while minimizing the BD-utilization. The CMC-LP alone, with a distance threshold of 100 miles, also achieves 100% of the demand, minimizing the BD-utilization. The speedup at light loads (for the CMC-(100) versus CMF-CMC(8000)) is a factor of 30.

Table 2 illustrate the results of the OSPF and KSP algorithms for the high-load scenario. OSPF(DEL) performs well, routing about 96.9% of the demand. However, the BD-utilization is sub-optimal, about 20% higher than optimal, and the average edge-load is sub-optimal, also about 11% higher than optimal. At heavy loads, OSPF is clearly unable to realize the *Max-Flow Min-Cost* routing. At heavier loads, the performance of OSPF degrades even further. KSP(16) achieves 99.4% of the demand, also with sub-optimal BD-utilization and average edge load. (The BD-utilization and edge loads can be improved by adding a Min-Cost LP after the KSP-LP.)

Table 2 also illustrates the results of the (i) the proposed CMF-CMC-LPs with distance constraints, and (iii) the proposed CMC-LP with distance constraints. (The EMMF-LP without distance constraints was intractable.) These LPs achieve $\geq$ 99% of the traffic demand, while achieving the minimal BD-utilization and edge loads given the distance constraints. The CMC-LP alone, with a distance threshold of 1000 miles, achieves 99.4% of the demand.

Table 3 illustrates the sizes of the LPs for the heavy-load scenario. The EMMF LP is intractable, as it requires $\approx$ 26K variables. The CMF-CMC LP with a distance threshold of 4000 miles is tractable, and achieves 99.6% of the demand. It requires $\approx$ 24K variables. The CMC LP with a distance threshold of 1000 miles is tractable, and achieves 99.4% of the demand. It requires $\approx$ 11K variables. The addition of cost constraints allows for a significant reduction in problem size, with a marginal reduction in the maximum flow rate.

We now explore the minimum-energy routing and BE-capacity of a network. Several different edge energy-cost models can be created. To illustrate the methodology, let the edge (i,j) energy cost reflect the router (i) power costs, using published data. According to [7], there are $\approx 10^6$ Internet routers each consuming $\approx$ 4 KW. Let a router of size 8x8 with 40 Gbps links consumes 4 KW = 4KJ/sec on average. The energy-cost of a line card (i.e., a fully-utilized IO port) in a router is therefore 500 W = 0.5 KJ/sec (comparable with the power of a Cisco CRS-1 linecard). According to [6], $\approx$ 30% of a router's power is due to IP packet processing, and this power can made to be proportional to the router utilization. Let α = 0.70 denote the fraction of a router's power which is constant (independent of utilization), and let $\beta = 1 - \alpha$ = 0.30 equal the fraction of power which is proportional to the router's utilization. According to published data, older routers are less energy-efficient than newer routers, typically by 33% per year or more. Therefore, let the energy-cost of an edge with utilization u equal $u \times \beta \times 0.5$KJ/sec.

An energy-efficient routing algorithm can use smallest-energy paths to route commodities. Several energy-costs were associated with the edges in E, with 1/3 edges having an energy-cost of $\beta \times 0.5$ KJ/sec, with 1/3 edges having 33% less efficiency, and 1/3 edges having 33% greater efficiency. Table 4 shows the routing results. OSPF(E) will route traffic along smallest-energy paths first. The CMF-CMC(x) LPs will find a

TABLE IV
MINIMUM-ENERGY-ROUTING, HEAVY-LOAD TRAFFIC MODEL

Algorithm	BE(D) (Gbjps)	BE(U) (Gbjps)	Flow percent	edge load (%)
OSPF (DEL)	312K	335K	95.3 %	65.9 %
OSPF (DIST)	”	337K	90.1 %	66.5 %
OSPF (HOP)	”	330K	93.0 %	64.8 %
OSPF (E)	”	327K	91.6%	66.2 %
CMF-CMC(4)	”	323K	98.6 %	64.8 %
CMF-CMC (2)	”	323K	98.3 %	64.7 %

Max-Flow with minimum energy-cost, with cost-threshold = $x \times \beta \times 0.5$ KJ/sec. According to table 4, the CMF-CMC(4) LP can achieve 98.6% of the flow demand, with near minimal energy requirements. In contrast, OSPF(E) achieves a lower flow rate of 91.6%, with higher energy costs and edge loads.

Current internet routers are not designed to save power at low utilizations, i.e., according to [7] they use $\approx$ 4 KW regardless of the load. In the near future, it is plausible to expect the router power to be largely proportional to the utilization, i.e., the fraction β should increase. For example, digital circuits can be clocked slower if they are under-utilized. In this case, the energy-savings due to minimum-energy routing can be significantly larger than shown in Table 4.

The proposed algorithms have been tested on numerous other network topologies and numerous other traffic patterns, and the results are consistent. The proposed LPs can result in considerably better energy-efficiencies, resource utilizations and edge loads than possible with other known routing algorithms, especially at higher loads. For large networks, the constrained LPs can reduce the size of the LPs to solve by factors of 10 or more, yielding tractable solutions to otherwise potentially intractable LPs.

VII. FUTURE INTERNET MODEL

In this section, we explore the use of the proposed routing algorithms to support Cloud-based web-services in a *Future Internet* network. The theory for a *Future Internet* network which can support 2 (or more) service classes, a new *Smooth* class and the usual *Best-Effort* class, has been presented in [26,27]. Legacy BE-Internet applications developed over the last 40 years typically use the TCP flow-control protocol, which results in quite bursty BE traffic flows [1-3]. These legacy applications will continue to run over the proposed *Future Internet*, using the same existing BE-Internet routing algorithms such as OSPF. New Cloud-based applications can be developed to exploit the new *Smooth* service class. It has been established in theory that the *Future Internet* network will provide the new *Smooth* class with *Deterministic and Essentially-Perfect* bandwidth, delay, jitter and QoS guarantees, for all *admissible* traffic demands within the *Capacity Region* of the network [26,27].

A *smooth* traffic flow is defined as a traffic flow between an (s,d) pair which exhibits a low burstiness or jitter, i.e., the packets arrive at a relatively smooth rate. *Essentially-Perfect* service for a smooth traffic flow is defined as service which deviates from perfect-service by at most K maximum-size packets, at any router and at any point in time. Formal definitions can be found in [26] and references therein.

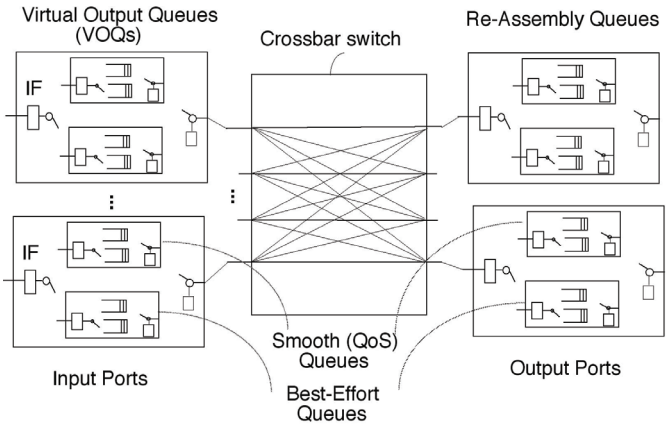

Fig. 4. Future Internet Router design.

In the proposed *Future Internet* network, the majority of buffering for the *Smooth* traffic class is moved *external to the network*, and the end-to-end queuing delays within the network for the *Smooth* traffic class are typically negligible relative to the fiber transmission delays [26]. The *Smooth* traffic flows can be viewed as *highly-efficient low-latency circuit-switched connections*, with deterministic QoS guarantees. Each Cloud data center has a token-bucket-based *Traffic Shaper Queue* (TSQ) to shape bursty QoS traffic into low-jitter streams before transmission. Each destination node has *Traffic Playback Queue* (TPQ) to regenerate the original bursty QoS stream(s) at the destinations with essentially-perfect QoS.

A *Future-Internet* router using an Input-Queued (IQ) switch is shown in Fig. 4. A router of size $M \times M$ consists of M input and output ports. Each input port has 2 classes of *Virtual Output Queues* (VOQs), the *Smooth* VOQs, and the BE VOQs. Each input port has a Input-Filter (IF) module to filter incoming packets and forward them to the appropriate VOQs. BE packets are forwarded to the BE-VOQs, while *Smooth* packets are forwarded to the Smooth-VOQs. The routing and scheduling of BE packets through the router is accomplished with the existing BE scheduling and routing algorithms (i.e., OSPF). The scheduling of *Smooth* packets through the router can be accomplished using deterministic schedules, which can be precomputed by the router when the smooth traffic flows are routed through the network [26].

The routing of *Smooth* commodity flows through a network can be centralized or distributed. The existing BE-Internet already supports a *Source-Routing* (SR) option, where a source can precompute an end-to-end path according to its own routing criterion, and prepend the routing information in the packet header. The existing BE-Internet also supports a *Policy-Based-Routing* (PBR) option, where a router can forward a packet to a particular output port based on several criterion in the packet header. Existing VOIP web-services already use proprietary QoS routing algorithms with these routing options, thereby bypassing the existing BE-Internet routing algorithms such as OSPF. In a centralized scheme, a *Traffic Engineering Routing* entity can maintain traffic demand and link-state matrices over time, perform centralized routing for high-

volume Cloud-based web-services, and distribute the routing information to the web-servers and routers. In a distributed scheme, each Cloud-based web-server can maintain traffic demand and link-state matrices using link-state distribution protocols, and perform SR or PBR. In both cases the packets in a *Smooth* flow will follow the precomputed path(s), using the SR or PBR options in the existing BE-Internet. Using the PBR option, each router is configured to identify *Smooth* packets from the the IP packet headers, and forward these to the appropriate Smooth-VOQs.

Current BE-Internet routers use heuristic *Best-Effort* schedulers, which cannot achieve deterministic bandwidth or QoS guarantees. Heuristic BE schedulers can typically achieve at best 80% link-efficiencies, and require very large packet buffers per input port. Current BE-Internet routers using heuristic BE schedulers typically use the classic *Delay-Bandwidth* buffer-sizing rule to determine router buffer sizes. A BE-Internet router with link speeds of 40 Gbps and a round-trip-time of 250 milliseconds typically requires buffers of $\approx$ 10 Gbits per input port, or $\approx$ 1 million maximum-size IP packets per input port [29]. To provide low delays, existing BE routers carrying real-time traffic are typically *over-provisioned* and operate at a small fraction (i.e., 33%) of peak capacity and link utilization [1-3]. This over-provisioning represents a significant loss of capacity, a significant energy cost, and a significant capital cost to pay for under-utilized capacity.

In the proposed *Future Internet*, no new VOQs or buffers are required, as existing BE-Internet routers already have very large buffers. The only new hardware required to support the *Smooth* traffic class in a router is the *Look Up Tables* (LUTs) to hold the precomputed schedules, which can easily fit on a small FPGA per linecard. The change in router design allows for the new *Smooth* service class to co-exist with the regular BE service class. All existing BE-Internet applications developed over the last 40 years will continue to work without any software changes over the proposed *Future Internet*. New applications which require improved throughput, energy-efficiency or QoS guarantees can be written to exploit the new highly-efficient *Smooth* service class.

A. Video Multicasting in the Future Internet

In this section, the delivery of aggregated video streams from Cloud-based web-servers over the proposed *Future Internet* is summarized. According to [8], video traffic from Netflix service represents $\approx$ 30% of the internet traffic in the USA during peak hours. Cloud-based data centers typically contain 50,000 individual servers in one location [8], and service thousands to millions of users.

Several prior papers have explored the aggregation of video traffic on the Internet. Ref. [30] explored the statistical multiplexing gain achievable when aggregating between 100 and 2,000 video streams in a smoothening buffer at the source. Ref. [31] showed that poor QoS of the BE-Internet can cause considerable problems for video distribution. Ref. [33] explored aggregating 100 video streams at the source using bufferless multiplexing. Ref. [34] explored aggregating 75 video streams at the source using a token-bucket traffic shaper queue. Our study differs from the prior art in 2 ways: (i) we explore the aggregation of $10^3 - 10^6$ video streams at one source, reflecting the glowing importance of Cloud data centers; (ii) we assume a *Future Internet* network which can provide deterministic QoS guarantees, as established in [26,27].

TABLE V
END-TO-END DELAY BOUNDS FOR AGGREGATED H.264/AVC VIDEO STREAMS.

Channels	EXBW	Hops	RQ Delay (millisec)	TSQ Delay (millisec)	TPQ Delay (millisec)
1	100%	10	83	8.9 sec	4.9 sec
10	50%	10	11	168	128
100	15%	10	1.4	25	37
1,000	5%	10	0.15	5.7	33
10,000	2%	10	≤ 0.1	1.5	33
10^5	1%	10	≤ 0.1	≤ 1	≤ 1
10^6	1%	10	≤ 0.1	≤ 1	≤ 1

Two real High-Definition (HD) video traces are used in our simulations, the *KAET Talk-Show* and the *Mars-to-China* traces available from the University of Arizona [32]. Both traces are 30-minute HD 1920x1080 video traces in the H.264/AVC format, with $\approx$ 51K frames (at 30 frames per second). The traces are self-similar and bursty. To generate an aggregated video stream to be delivered, copies of each 1/2 hour video stream are circularly rotated by a random amount and added together yielding an aggregated 1/2 hour video stream as described in prior papers [26,27,30-35]. (No processing of the video streams to minimize the jitter of the aggregated stream is assumed.)

Assume that a cloud-based web-server will use the SR or PBR options to traverse path(s) with a specified rate, and transport highly aggregated video over the path(s). The number of aggregated video streams on each path is known in advance, and the *minimum-bandwidth* required by the path is also known. As described in [26,27], the cloud-server will generate an aggregate smoothed stream with an additional *excess-bandwidth* component, typically between 1% and 5% of the *minimum-bandwidth* requirement, to control the TSQ and TPQ queueing delays.

To find the total delays to deliver an aggregated video stream, the TSQ at the source and the TPQ at the destination were simulated using the methodology described in [26,27,30-35]. Fig. 5 illustrates the application-layer queueing delays in the TSQ and the TPQ for the aggregation of between A =1...1M individual video streams. For each point in Fig. 5, the TSQ and TPQ were simulated with 100 randomly generated aggregated video streams, each 1/2 hour in length. The x-axis illustrates the excess-bandwidth provisioned in the path(s). The y-axis illustrates the mean queueing delays. The 95% confidence intervals are very small. The application-layer queueing delays in the TSQ and TPQ drop rapidly as the excess-bandwidth increases. (The TSQ and TPQ can easily be incorporated into a software multicast overlay layer.)

Table 5 illustrates the end-to-end router queueing delays (RQ) for aggregated video streams traversing 10 routers (based on the KAET Talk Show video trace). Each row represents a level of aggregation and excess-bandwidth in the provisioned path (or tree). The aggregation of 1,000 video streams each requiring 1.464 Mbits per second (Mbps) will require a mini-

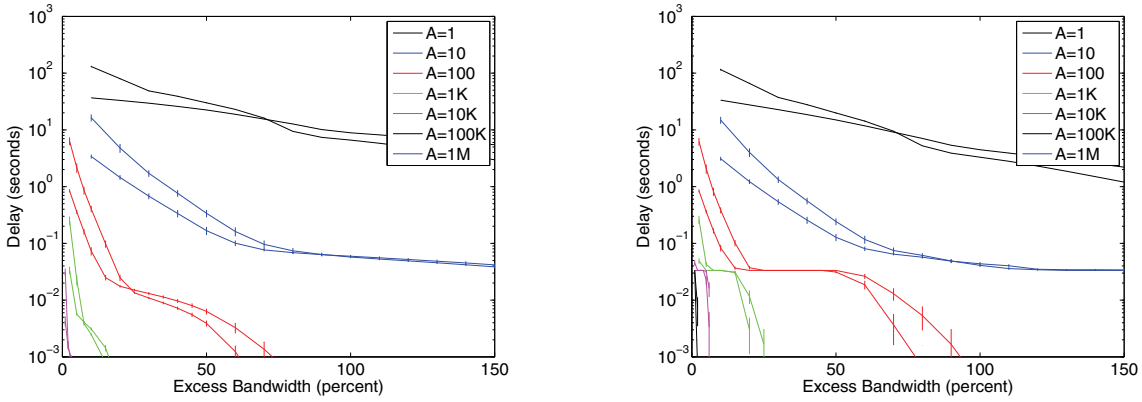

Fig. 5. (a) TSQ delay vs. excess bandwidth. (b) TPQ delay vs. excess bandwidth.

mum aggregate bandwidth of 1.464 Gbps. To achieve a small queueing delay in the TSQ and TPQ, let an excess bandwidth of 5% be used in the path (or tree), so the provisioned rate is 1.05x1.464 = 1.537 Gbps.

For an aggregation of 1,000 video streams with an excess-bandwidth of 5%, the mean end-to-end router queueing delay is $\leq$ 1 millisec, which is extremely low compared to the existing BE-Internet. The mean queueing delays in the TSQ and TPQ are also very small. The bandwidth-efficiency and resource-efficiency of this path (or tree) is 95%, far higher than possible in the existing BE-Internet. Each router buffers on average $\leq$ 2 packets per QoS-enabled traffic flow, several orders of magnitude less buffering than for BE traffic flows [26]. It is estimated that router buffers represent a reasonable fraction of the cost, size and power dissipation of existing Internet routers.

The same methodology has been tested on hundreds of other self-similar streams. The TSQ and TPQ queueing delays are always consistent with Fig. 5; High levels of aggregation ($\geq$ 1,000 streams) enables multiple self-similar video streams to be shaped into a sufficiently smooth traffic flow at the source node. The smoothened traffic flows can be routed by the *Max-Flow-Min-Cost* algorithms presented in this paper to achieve improved throughput, energy-efficiency and QoS guarantees in the proposed *Future Internet*. The largest energy-savings are a result of removing the need to significantly over-provision the links. Using the *Smooth* traffic class, links can carry traffic at 95% - 100% loads and provide deterministic and essentially-perfect QoS guarantees. The aggregate throughput and energy-efficiency of the entire backbone network will improve correspondingly. In contrast, the BE-Internet operates links carrying time-sensitive traffic at light loads, typically $\leq$ 33% load, to achieve weak statistical QoS guarantees [1-3].

VIII. Conclusions

A *Constrained Multicommodity Max-Flow-Min-Cost* algorithm for routing unicast traffic flows in a network subject to cost constraints has been presented. The addition of routing cost constraints can result in significantly smaller LPs to solve, and can result in significantly better resource-utilization and edge-loads. When the cost constraints are relaxed, no other unicast routing algorithms can achieve larger *Maximum Flows*, or lower costs given the *Maximum-Flow* rates to be supported; These unicast routing algorithms can achieve the lowest energy-costs given the *Maximum-Flow* rates to be supported. These routing algorithms have polynomial-time solutions, in contrast to traditional multipath routing algorithms which can be NP-Hard. It is also shown that every network has a finite *Bandwidth-Cost* capacity which cannot be exceeded. Two capacities where explored, the *Bandwidth-Distance* capacity and the *Bandwidth-Energy* capacity. The proposed routing algorithms can achieve *Maximum-Flows* with minimal BD and BE costs, subject to cost constraints imposed by a network administrator. We also present some new insights into *Multicast Maximum-Flow-Minimum-Energy* routing in networks using *Network Coding*. It is shown that the energy costs of different multicast routings that support the same multicast flow rate can be significantly different. The application of these routing algorithms to route aggregated and smoothened video streams from Cloud data centers in a proposed *Future-Internet* network with improved throughput, energy-efficiency and QoS guarantees is presented.

A. Acknowledgements

The author acknowledges the insightful comments of the Editor and Reviewers, which have significantly improved the paper.

References

[1] L. G. Roberts, "A radical new router: the Internet is broken—let's fix it," *IEEE Spectrum*, July 2009.
[2] P. Gevros, J. Crowcroft, P. Kerstein, and S. Bhatti, "Congestion control mechanisms and the best-effort service model," *IEEE Network Mag.*, May/June 2001.
[3] V. Joseph and B. Chapman, *Deploying QoS for Cisco IP and Next-Generation Networks: The Definitive Guide*. Elsevier/Morgan-Kaufman Publishers, 2009.
[4] X. Xiao and L. M. Ni, "Internet QoS: a big picture," *IEEE Network Mag.*, Mar./Apr. 1999.
[5] A. Meddeb, "Internet QoS: pieces of the puzzle," *IEEE Commun. Mag.*, Jan. 2010.
[6] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future Internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Commun. Surveys and Tutorials*, second quarter, 2011.

[7] B. Raghavan and J. Ma, "The energy and emergy of the Internet," *Hotnets 2011*.

[8] J. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th edition. Elsevier/Morgan Kaufmann, 2011.

[9] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, July 2000.

[10] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *JACM*, vol. 37, no. 2, pp. 318–334, Apr. 1990.

[11] T. Leighton and S, Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *JACM*, vol. 46, no. 6, pp. 787–832, Nov. 1999.

[12] S. G. Kolliopoulos and C. Stein, "Improved approximation algorithms for unsplittable flow problems," in *Proc. 1997 IEEE FOCS*, pp. 426–436.

[13] V. Guruswam. S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," *J. Computer and System Sciences*, vol. 67, no. 3, pp. 473–496, Nov. 2003.

[14] S. Toumpis and A. J. Goldsmith, "Capacity regions for ad-hoc wireless networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, July 2003.

[15] P. Giaccone, "Throughput region of finite-buffered networks," *IEEE Trans. Parallel and Dist. Systems*, vol. 18, no. 2, pp. 251–263, Feb. 2007.

[16] M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "BoxRouter 2.0: architecture and implementation of a hybrid and robust global router," *2007 IEE/ACM Int. Conf. CAD*.

[17] C. Albrecht, "Global routing by new approximation algorithms for multicommodity flow," *IEEE Trans. CAD-ICS*, vol. 20, no. 5, pp. 622–632, May 2001.

[18] T. H. Szymanski, "Achieving minimum-routing-cost maximum-flows in infrastructure wireless mesh networks," *2012 IEEE WCNC*.

[19] M. Pioro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. Elsevier/Morgan-Kaufmann, 2004.

[20] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols and Architectures*. Elsevier/Morgan-Kaufmann, 2004.

[21] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, 2nd edition. McGraw-Hill, 2004.

[22] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke, "Optimal oblivious routing in polynomial time," *2003 ACM Symp. Theory of Computing*.

[23] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, June 2006.

[24] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, June 2006.

[25] Y. Xuan and C.-T. Lea, "Network-coding multicast networks with QoS guarantees," *IEEE Trans. Networking*, vol. 19, no. 1, Feb. 2011.

[26] T. H. Szymanski and D. Gilbert, "Provisioning mission-critical telerobotic control systems over Internet backbone networks with essentially-perfect QoS," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, June 2010.

[27] T. H. Szymanski, "Future Internet video multicasting with essentially perfect resource utilization and QoS guarantees," *2011 IEEE iWQoS Workshop*.

[28] S. Orlowski, R. Wessaly, M. Pioro, and A. Tomaszewski, "SNDlib 1.0–Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, Oct. 2009.

[29] S. Iyer, R. R. Kompella, and N. Mckeown, "Designing packet buffers for router linecards," *IEEE Trans. Networking*, vol. 16, no. 3, June 2008.

[30] Z.-L. Zhang, J. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing and call admission control for stored video," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1148–1166, 1997.

[31] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Trans. Circuits and Systems for Video Technol.*, vol. 11, no. 3, Mar. 2001

[32] P. Seelingm, M. Reisslein, and B. Kulapa, "Network performance evaluation using frame size and quality traces of single layer and two layer video," *IEEE Commun. Surveys and Tutorials*, vol. 6, no. 3, 2004.

[33] G. Van der Auwera and M. Reisslein, "Implications of smoothing on statistical multiplexing of H.264/AVC and SVC video streams," *IEEE Trans,. Broadcast.*, vol. 55, no. 3, pp. 541–558, Sept. 2009.

[34] T. H. Szymanski and D. Gilbert, "Internet multicasting of IPTV with essentially-zero delay jitter," *IEEE Trans. Broadcast.*, vol. 55, no. 1, Mar. 2009.

[35] S. K. Srinivasan, J. Vahabzadeh-Hagh, and M. Reisslein, "The effects of priority levels and buffering on the statistical multiplexing of single-layer H.264/AVC and SVC encoded video streams," *IEEE Trans. Broadcast.*, vol. 56, no. 3, Sept. 2010.

[36] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and J. Hu, "Efficient congestion mitigation using congestion-aware Steiner trees and network coding topologies," *VLSI Design*, vol. 2011, article ID 892310.

[37] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and J. Hu, "Efficient rerouting algorithm for congestion mitigation," in *Proc. 2009 IEEE Annual Symp. on VLSI*, pp. 43–48.

Ted H. Szymanski (M'87) completed a BaSc. in Engineering Science and the MaSc. and PhD degrees in Electrical Engineering at the University of Toronto. He has held faculty positions at Columbia University, where he was affiliated with the NSF-funded Center for Telecommunications Research (CTR), and McGill University, where he was affiliated with the Canadian Institute for Telecommunications Research (CITR). From 1993 to 2003, he led the 'Optical Architectures' project in a Canadian national research program on Photonic Systems funded by the Networks of Centers of Excellence (NCE) program. The program brought together significant industrial and academic collaborators, including Nortel Networks (now Ericson), Newbridge Networks (now Alcatel), Lockheed-Martin/Sanders, Lucent Technologies and McGill, McMaster, Toronto and Heriot-Watt Universities. The program demonstrated a free-space "intelligent optical backplane" architecture exploiting emerging optoelectronic smart-pixel technologies, with approx. 1K smart pixels and microscopic laser channels per square centimeter of bisection area. He holds two patents related to this project, the first on "intelligent optical networks" using smart-pixel arrays, and the second on embedded Forward-Error-Correction to improve throughput in smart-pixel based optical networks. He also holds a patent on low-jitter scheduling techniques to achieve improved QoS guarantees in Internet routers. From 2001-2011, he held the Red Wilson / Bell Canada Chair in Data Communications at McMaster University, where he has also served as the Associate Chair (undergraduate) and the undergraduate student advisor in the ECE Department. After the decline of the Canadian optical networking industry during 2000 - 2005, he expanded his research interests to include the Future Internet and wireless architectures, and improved energy-efficiency, resource-utilization and QoS guarantees for emerging Cloud-based services.