Efficient and Distributed Computation of Maximum Multicast Rates

Zongpeng Li, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{arcane, bli}@eecg.toronto.edu

***Abstract*— The transmission of information within a data network is constrained by network topology and link capacities. In this paper, we study the fundamental upper bound of information multicast rates with these constraints, given the unique replicable and encodable property of information flows. Based on recent information theory advances in coded multicast rates, we are able to formulate the maximum multicast rate problem as a linear network optimization problem, assuming the general undirected network model. We then proceed to apply Lagrangian relaxation techniques to obtain (1) a necessary and sufficient condition for multicast rate feasibility, and (2) a subgradient solution for computing the maximum rate and the optimal routing strategy to achieve it. The condition we give is a generalization of the well-known conditions for the unicast and broadcast cases. Our subgradient solution takes advantage of the underlying network flow structure of the problem, and therefore outperforms general linear programming solving techniques. It also admits a natural intuitive interpretation, and is amenable to fully distributed implementations.**

Index Terms: Graph Theory, Information Theory, Mathematical Programming/Optimization.

I. Introduction

Packet transmission in data networks may be modeled as the flow of bit streams, referred to as information flows. Compared to classical network flows, *e.g.*, fluid flows in a pipe network, information flows share some common fundamental properties while differ in others. As compared in the table below, both fluid flow and information flow need to confine to the network topology, and to respect link capacities. However, information flows may be replicated or encoded, while fluid flows may not.

	topology confinement	respect link cap.	replicable	encodable
info. flow	yes	yes	yes	yes
fluid flow	yes	yes	no	no

Replication and encoding have been shown to be fundamental capabilities in achieving high information multicast rates [1], [2], [3]. Multicast refers to the form of one-to-many data transmission. Ahlswede *et al.* and Koetter *et al.* recently proved that, in a directed network, a multicast rate χ is feasible if and only if the max-flow rate from the source to each receiver is at least χ [1], [2]. Fig. 1 illustrates the concept of multicast rate, data replication and data encoding with two simple multicast transmissions. All links in the examples have a unit capacity of 1 bit per second.

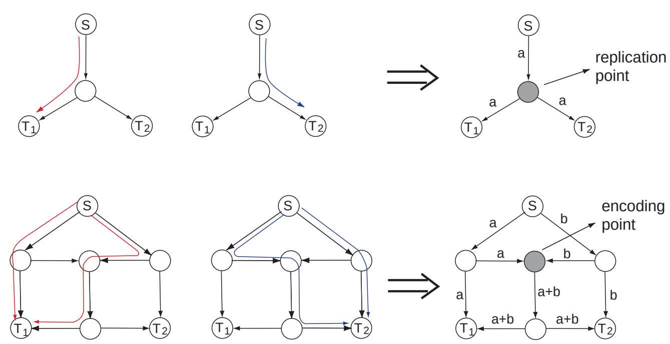

Fig. 1. Multicast rates with information replication and information encoding.

In the first example, the max-flow rate from S to either T_1 or T_2 is 1 bit per second. Although these two flows share their first link, It is still possible to multicast information to both T_1 and T_2 at the rate of 1 bit per second. We can send just one copy of the flow at the shared link, then replicate it into two identical copies and forward one along each downstream link.

In the second example, the max-flow rate from S to either T_1 or T_2 is 2 bits per second. Again these two flows share a common relay link in the middle of the network. This time the problem can be resolved by data encoding. Two distinct unit rate incoming flows, a and

b, are encoded at the tail of the shared link, using the $+$ operation defined in the Galois field. The encoded flow, which also has a unit rate of 1 bit per second, is then sent through the shared link to its head, where the encoded flow is replicated and further routed to both receivers. Each receiver can recover the two original flows a and b from the two flows they receive.

For directed networks, the aforementioned result due to Ahlswede *et al.* and Koetter *et al.* constitutes a nice necessary and sufficient condition for multicast rate feasibility. It further leads to an efficient solution for computing the maximum multicast rate. One just needs to compute a maximum flow from the multicast sender to each receiver, independently. Then the minimum of these max-flow rates is the maximum achievable rate for the entire multicast connection.

However, these results do not directly apply in the more general undirected network model [4], for which no non-trivial necessary and sufficient feasibility condition is known, and no efficient algorithm for computing the maximum rate has been proposed. In an undirected network model, each link is bi-directional, and flows in both directions share its capacity. The study of the undirected network model is supported by the following justifications. First, as past research in network flow theory [5] and information theory [4] suggests, the undirected network model has its own rhythm, and results obtained there may be drastically different from those obtained in the directed network model. In fact, the undirected model is more general in that, a solution constructed for undirected networks can usually be applied to solve the same problem in directed networks, but not vice versa. This is particularly true for our problem and solution in this paper. Second, undirected links provide the complete flexibility in capacity allocation, and consequently leads to higher transmission rates that better represent the optimal information delivery rate, compared to static link capacity allocation in directed networks. Finally, in special network scenarios such as wireless ad hoc networks, the communication link is naturally undirected, in the sense that data transmission along both directions of the wireless link share the available spectrum [6], [7].

In this paper, we study the maximum multicast rate problem in undirected networks. Our objectives include both a necessary and sufficient feasibility condition, and an efficient, distributed algorithm to compute the maximum multicast rate. Towards this direction, we first formulate the maximum multicast rate problem into a linear network optimization problem. We present and interpret both the primal and dual linear programs, on each of which we then apply Lagrangian relaxation techniques. Lagrangian dualization has been proven to be an effective method both for deriving max-min style iff conditions [5], and for designing efficient distributed solutions for convex optimization problems [8], [9].

The necessary and sufficient condition we derive in this paper is a generalization of the well known conditions for the unicast and broadcast cases, and is obtained through studies of the Lagrangian dual of the primal linear program. Applying Lagrangian relaxation on the dual program leads to a subgradient solution, which has an appealing intuitive interpretation: it iteratively improves an existing orientation of the original network based on the link saturation level, until an optimal one is reached. Then a number of maximum flow computations are invoked to determine the maximum flow rate and the corresponding flow routing strategy. Our algorithm takes advantage of the underlying network flow structure of the problem, and consists of mostly max-flow/min-cut computations. It outperforms general solution techniques such as the simplex method, which solves the linear program as a black-box and ignores its background. We also show that our algorithm allows a fully distributed implementation.

The rest of the paper is organized as follows. We present related work in Sec. II, give linear programming formulations of our problem in Sec. III, derive the iff condition in Sec. IV, and construct the subgradient algorithm in Sec. V. We then conclude the paper in Sec. VI.

II. Related Work

Recent research in information theory discovers that routing alone is not sufficient to achieve maximum information transmission rate across a data network [1], [2]. Rather, applying encoding and decoding operations at relay nodes as well as at the sender and receivers, are in general necessary in an optimal transmission strategy. Such coding operations are referred to as *network coding*. The pioneering work by Ahlswede *et al.* [1] and Koetter *et al.* [2] proves that, in a directed network with network coding support, a multicast rate is feasible if and only if it is feasible for a unicast from the sender to each receiver. Li *et al.* [10] then prove that linear coding usually suffices in achieving the maximum rate.

In [11], Sanders *et al.* study efficient code assignment in directed acyclic networks. They design polynomial time algorithms that determine the coding operations to be applied at each node, in order to achieve the maximum multicast rate. Their result improves the previous algorithm of Li *et al.*, which performs exponentially many linear independence inspections [10]. Code assignment is complementary to our work in this paper. Our subgradient algorithm finds the optimal routing strategy, which specifies how much flow is to be routed through each link. Code assignment then determines the content of these flows, *i.e.*, their linear relation with the original information flows at the sender.

In [3], we show that for undirected networks, the potential of network coding to improve multicast rate is rather limited: bounded by a factor of 2 in theory (the bound 2 is for the fractional case; in the integral case, the best known bound is 26 implied by Lau's recent work [12]), and usually much smaller in practice. However, the introduction of network coding dramatically reduces the computational complexity of finding the maximum multicast rate and the strategy to achieve it. Without network coding, the maximum multicast rate problem is equivalent to Steiner tree packing, which is a well-known NP-complete problem [13]. With network coding, the maximum multicast rate can be computed via linear optimization. In [4], we show that for either a single communication session or multiple sessions, either with network coding support at all nodes or at edge nodes only, the maximum transmission rate problem can be formulated into linear network optimization. While [4] gives only the primal linear program without a solution method, we work on both the primal and dual linear programs in this paper, and provide an efficient, distributed solution.

On the application side, network coding research has spawned a number of coded multicast system design recently. These systems are usually built upon application-layer overlay networks or wireless ad hoc networks, where each node is a fully-functional host, and therefore possesses data encoding capabilities. These systems differ from previous multicast protocols in that information is no longer transmitted along a single multicast tree, or a collection of multicast trees. For instance, Zhu *et al.* utilize network coding to build k-redundant multicast mesh in application-layer overlay networks [14]. Chou *et al.* design robust de-centralized network coding schemes for broadcast transmission [15]. Wang *et al.* [16] completed a real-world coded multicast implementation targeting near-optimal throughput. It includes an ϵ-relaxation based algorithm for computing the maximum-rate multicast topology, a randomized code assignment component, and a coding library that supports network coding operations over finite field $GF(2^8)$ or $GF(2^{16})$.

Traditional network flow theory studies the transmission of goods within a capacitied transportation network. The maximum transmission rate between two nodes is characterized by the celebrated max-flow min-cut theorem [5]: *a flow rate χ between nodes u and v is feasible, if and only if every cut between u and v has size at least χ.* Various algorithms may compute the maximum flow efficiently, some of which allow fully distributed implementation, *e.g.*, the push-relabel algorithm [5] and the ϵ-relaxation algorithm [8].

Our work in this paper was also inspired by a preliminary version of [9], in which Lun *et al.* successfully design subgradient algorithms for computing the min-cost multicast topology in directed networks. Both their algorithm and ours target efficiency and potential for distributed implementation. Their algorithm works on a partial Lagrangian dual of the primal problem, and employs primal recovery techniques to obtain the entire optimal solution. Our algorithm applies Lagrangian relaxation on the dual problem, and compute the entire optimal primal solution from partial primal solution through pure combinatorial computations.

III. Maximum multicast rate: linear programming formulation

In [4], we have given the primal linear program for the maximum multicast rate problem in undirected networks. Here we present this LP again for completeness. We also give the dual program, which will be used in the design of our subgradient algorithm in Sec.V. Our primal and dual linear programs have an underlying structure of network flow and cut, respectively. For the ease of understanding and later reference, we present the max-flow and min-cut linear programs first.

A. The max-flow LP and the min-cut LP

Let $G = (V, E)$ be the network topology, and the constant vector $C \in Q_+^E$ be capacities of the undirected links, where Q_+ denotes the set of nonnegative rational numbers. In the max-flow LP, $\overrightarrow{TS}$ is a directed virtual link with infinite capacity, going from the destination T to the source S. $N(u)$ denotes the set of neighbors of node u. $f \in Q_+^A$ is the flow vector, where $A = \{\overrightarrow{uv}, \overrightarrow{vu} \;|uv \in E\}$ is the set of directed arcs. The scalar χ is the overall end-to-end flow rate. The max-flow LP essentially maximizes the end-to-end flow rate, with link capacity limits and flow conservation requirements (total incoming flow rate at a node equals its total outgoing flow rate). Flow conservation at source and destination nodes are possible due to the virtual link $\overrightarrow{TS}$ we add, the flow rate on which exactly equals the overall flow rate from S to T.

The max-flow linear program

Maximize $\quad \chi = f(\overrightarrow{TS})$

Subject to:

$$\begin{cases} f(\overrightarrow{uv}) \leq C(uv) & \forall\, \overrightarrow{uv} \neq \overrightarrow{TS} \\ \sum_{v\in N(u)} f(\overrightarrow{uv}) = \sum_{v\in N(u)} f(\overrightarrow{vu}) & \forall u \end{cases}$$

$$f(\overrightarrow{uv}) \geq 0 \qquad \forall\, \overrightarrow{uv}$$

The min-cut linear program

Minimize $\quad \sum_{\overrightarrow{uv}} C(uv) y(\overrightarrow{uv})$

Subject to:

$$\begin{cases} y(\overrightarrow{uv}) + p(v) \geq p(u) & \forall\, \overrightarrow{uv} \neq \overrightarrow{TS} \\ p(T) - p(S) \geq 1 \end{cases}$$

$$y(\overrightarrow{uv}) \geq 0 \qquad \forall\, \overrightarrow{uv}$$

In the min-cut LP, vector y indicates which links are "cut". This LP always has an optimal solution that is integral, where each entry in y is valued to either 1 or 0, indicating whether the corresponding link is in the min-cut or not. The constraints imply that, for each path P connecting the source S to the destination T, $\sum_{\overrightarrow{uv}\in P} y_i \geq 1$, *i.e.*, at least one link along the path is cut. The objective is to minimize the total link capacity being cut.

B. The primal linear program

In the primal LP for the maximum multicast rate problem, vector $c : Q_+^A$ stores capacities for directed links, *i.e.*, the allocation of the undirected link capacity in both directions. The sender node is S, and the receiver nodes are $T_1, \ldots, T_k$. χ is the overall multicast rate. Vectors $f_i \in Q_+^A$ denotes a network flow from sender S to each receiver T_i. Directed links $\overrightarrow{T_iS}$ with infinite capacity are again introduced for a concise presentation of the LP.

Constraints in the primal program require capacities allocated to both directions not to exceed the undirected link capacity (4), each flow f_i to be a valid network flow (2)(3), and the multicast rate not to exceed any of these network flow rate (1). Essentially, the primal LP tries to establish an orientation of the undirected network, within which to set up independent network flows from

Maximize $\quad \chi$

Subject to:

$$\begin{cases} \chi \leq f_i(\overrightarrow{T_iS}) & \forall i & (1) \\ f_i(\overrightarrow{uv}) \leq c(\overrightarrow{uv}) & \forall i, \forall\, \overrightarrow{uv} \neq \overrightarrow{T_iS} & (2) \\ \sum_{v\in N(u)} f_i(\overrightarrow{uv}) = \sum_{v\in N(u)} f_i(\overrightarrow{vu}) & \forall i, \forall u & (3) \\ c(\overrightarrow{uv}) + c(\overrightarrow{vu}) \leq C(uv) & \forall uv \neq T_iS & (4) \end{cases}$$

$$c(\overrightarrow{uv}), f_i(\overrightarrow{uv}), \chi \geq 0 \qquad \forall i, \forall\, \overrightarrow{uv}$$

the sender S to each receiver T_i — and do so in an optimal way, in that the minimum of the independent max-flow rates — which by the result of Ahlswede *et al.* [1] equals to the multicast rate — is maximized. A feasible solution to the primal LP provides an orientation of the original network, $c(\overrightarrow{uv})$; a flow routing scheme, $f(\overrightarrow{uv}) = \max_i f_i(\overrightarrow{uv})$; and a feasible multicast rate, χ.

C. The dual linear program

The dual linear program for the maximum multicast rate problem is:

Minimize $\quad \sum_{uv} C(uv)x(uv)$

Subject to:

$$\begin{cases} x(uv) \geq \sum_i y_i(\overrightarrow{uv}) & \forall uv \neq T_iS & (5) \\ y_i(\overrightarrow{uv}) + p_i(v) \geq p_i(u) & \forall i, \forall\, \overrightarrow{uv} \neq \overrightarrow{T_iS} & (6) \\ p_i(T_i) - p_i(S) \geq z_i & \forall i & (7) \\ \sum_i z_i \geq 1 & & (8) \end{cases}$$

$$x(uv), y_i(\overrightarrow{uv}), z_i \geq 0 \qquad \forall i, \forall\, \overrightarrow{uv}$$

While the primal LP is in the form of flow maximization, the dual LP is in the form of cut minimization. In an optimal solution, each dual variable in vectors x, y and z is valued between 0 and 1. In the dual constraints, (8) distributes weights among the cuts between S and each T_i. (6) and (7) require each cut y_i to be a valid cut, except that an edge in the cut will now be cut to percentage z_i, rather than 100% as in the minimum cut LP. Then the cut values of a link in the k different cuts are added up in (5). If the summations in both directions differ, the larger one is taken to be the cut value for the undirected link.

The variable-constraint correspondence in the primal and dual LPs is given in the table below. It will later help us decide which constraints to relax.

primal	(1)	(2)	(3)	(4)	c	$f(\overrightarrow{uv})$	$f(\overrightarrow{T_iS})$	χ
dual	z	y	p	x	(5)	(6)	(7)	(8)

D. Performance of general LP solvers

Both the primal and the dual LPs have $O(km)$ number of variables and $O(km)$ number of constraints, where k is the number of multicast receivers, and $m = |E|$ is the number of links in the network. Since linear programming is polynomial time solvable in general, it follows that the maximum multicast rate can be computed in polynomial time, even for undirected networks.

However, experiences show that for network flow type problems with extra side constraints, *e.g.*, the multicommodity flow problem, the performance of general linear programming techniques are often below acceptable levels, when the size of the problem is relatively large. For the multicast rate problem in particular, we have experimented with both the simplex method and the primal-dual interior-point method, as implemented in `glpk 4.4` [17]. We apply both methods to solve the primal LP as a black-box, on networks and multicast groups with various sizes. Our findings show that, on a typical Pentium IV computing platform, the interior-point method may handle networks with a few thousand links within a reasonable amount of time (on the order of seconds), as long as the multicast group is small ($k \leq 5$). For networks that are larger, or for a broadcast network with a few hundred nodes and less than one thousand links, the computation easily takes hours. The performance of the simplex method is constantly worse than that of the interior-point method.

Another critical drawback of applying general linear programming methods, is that these methods are inherently centralized, requiring global information being collected to one central point of computation. The solution we construct in Sec. V solves both problems. It decomposes the maximum multicast rate computation into a sequence of max-flow/min-cut computations, for which very efficient algorithms exist and can be applied. It also allows the computation to be distributed onto each node in the network, where only local information is collected.

IV. MULTICAST RATE FEASIBILITY: THE NECESSARY AND SUFFICIENT CONDITION

We now apply Lagrangian relaxation on the primal LP to derive the necessary and sufficient condition for multicast rate feasibility in undirected networks. We explain how it generalizes the conditions in unicast and broadcast cases, and provide an interpretation from the perspective of bandwidth efficiency.

A. The condition as a theorem

Theorem 1. A multicast rate χ is feasible in an undirected network G, if and only if for every link distance function $x \in Q_+^E$,

$$\frac{|G|_x}{Min_{\chi(f)=1}|f|_x} \geq \chi.$$

In the theorem above, $|G|_x$ denotes the size of the network under distance vector x, *i.e.*, $|G|_x = \sum_{uv} C(uv)x(uv)$. $f \in Q_+^A$ denotes a multicast topology, or a flow routing scheme; and $|f|_x = \sum_{\overrightarrow{uv}} f(\overrightarrow{uv})x(uv)$ is the size of the multicast topology, under distance vector x. $\mathrm{Min}_{\chi(f)=1}|f|_x$ denotes the size of the minimum multicast topology that achieves unit multicast rate. Note a multicast topology is not necessarily a multicast tree — the second multicast transmission in Fig. 1 constitutes a counter example.

B. The proof of correctness

Proof of Theorem 1: Consider the primal multicast rate LP given in Sec. III-B. We now formulate its Lagrangian dual by relaxing the undirected link capacity constraints (4), and introduce corresponding prices into the objective function, which becomes:

$$\chi - \sum_{uv} x(uv)\Delta(uv).$$

In the modified objective function above, $\Delta(uv) = c(\overrightarrow{uv}) + c(\overrightarrow{vu}) - C(uv)$ denotes the amount of capacity over-use at link uv, and $x(uv)$ is the Lagrangian multiplier acting as the unit price charged for capacity over-use. At this point, the primal multicast rate LP is transferred into the Lagrangian subproblem:

$$L(x) = \mathrm{Max}_P[\chi - \sum_{uv} x(uv)\Delta(uv)],$$

with P being the following polytope:

$$P: \begin{cases} \chi \leq f_i(\overrightarrow{T_iS}) & \forall i \\ f_i(\overrightarrow{uv}) \leq c(\overrightarrow{uv}) & \forall i, \forall\ \overrightarrow{uv} \neq \overrightarrow{T_iS} \\ \sum_{v \in N(u)} f_i(\overrightarrow{uv}) = \sum_{v \in N(u)} f_i(\overrightarrow{vu}) & \forall i, \forall u \\ c(\overrightarrow{uv}), f_i(\overrightarrow{uv}), \chi \geq 0 & \forall i, \forall\ \overrightarrow{uv} \end{cases}$$

The Lagrangian dual problem is then:

The Lagrangian duality theorem assures that each feasible value of $L(x)$ is an upper-bound for a feasible

Minimize $\quad L(x)$
Subject to: $\quad x \geq 0$

multicast rates χ. Furthermore, this bound is tight in the sense that the minimum value of $L(x)$ exactly matches the maximum achievable rate χ, *i.e.*, the optimal objective values of the primal LP and the Lagrangian dual are equal. Consequently, the maximum multicast rate χ^* can be computed as:

$$\chi^* = \text{Min}_{x\geq 0}\{\text{Max}_P[\chi - \sum_{uv} x(uv)\Delta(uv)]\}$$

We now perform manipulations on the expression of χ^*, and provide justifications for each step.

$$\begin{aligned}
&\chi^* \\
=_1 \quad & \text{Min}_{x\geq 0}\{\text{Max}_P[\chi - \textstyle\sum_{uv} x(uv)\Delta(uv)]\} \\
=_2 \quad & \text{Min}_{x\geq 0}\{\text{Max}_P[\chi - \textstyle\sum_{\vec{uv}} x(uv)c(\vec{uv}) \\
& + \textstyle\sum_{uv} x(uv)C(uv)]\} \\
=_3 \quad & \text{Min}_{x\geq 0}\{\text{Max}_P[\chi - |f|_x + |G|_x]\} \\
=_4 \quad & \text{Min}_{x\geq 0,\text{Min}_{\chi(f)=1}|f|_x\geq 1}\{\text{Max}_P[\chi - |f|_x + |G|_x]\} \\
=_5 \quad & \text{Min}_{x\geq 0,\text{Min}_{\chi(f)=1}|f|_x\geq 1}|G|_x \\
=_6 \quad & \text{Min}_{x\geq 0,\text{Min}_{\chi(f)=1}|f|_x= 1}|G|_x \\
=_7 \quad & \text{Min}_{x\geq 0}\frac{|G|_x}{\text{Min}_{\chi(f)=1}|f|_x}
\end{aligned}$$

In the derivations above, $=_1$ holds due to Lagrangian duality, as discussed early. $=_2$ and $=_3$ are due to definitions. $=_4$ is due to dual feasibility. The inner maximization subproblem is unbounded in cases where $\text{Min}_{\chi(f)=1}|f|_x < 1$ — one may scale up flows in f to arbitrarily large, and hence scaling up the difference between χ and $|f|_x$ to arbitrarily large. $=_5$ is due to the fact that when $\text{Min}_{\chi(f)=1}|f|_x \geq 1$, we have $\chi - |f|_x \leq 0$, and $\text{Max}_P[\chi - |f|_x + |G|_x] = |G|_x$. $=_6$ is due to the observation that for every x where $\text{Min}_{\chi(f)=1}|f|_x > 1$, there exists another vector $x' = x/\text{Min}_{\chi(f)=1}|f|_x$, such that $\text{Min}_{\chi(f)=1}|f|_{x'} = 1$, and $|G|_{x'} < |G|_x$. Finally, $=_7$ is due to the fact that if we scale link distances in x proportionally, the ratio $|G|_x/|f|_x$ remains at the same value.

Now we can claim $\chi^* = \text{Min}_{x\geq 0}\frac{|G|_x}{\text{Min}_{\chi(f)=1}|f|_x}$, and that concludes the proof of Theorem 1. □

C. Interpretation and discussions

Comparison with unicast and broadcast cases

A unicast is an one-to-one data transmission, and a broadcast is an one-to-all data transmission. It is known that for unicast or broadcast, encodability does not make a difference in the maximum achievable transmission rate [3]. Therefore, each atomic unicast topology is a path, and each atomic broadcast topology is a spanning tree. The maximum unicast rate problem is equivalent to the path packing or maximum flow problem, and the maximum broadcast rate problem is equivalent to the spanning tree packing problem. For unicast rate feasibility, the max-flow min-cut theorem constitutes an elegant necessary and sufficient condition. For broadcast rate feasibility, Tutte-Nash-Williams' theorem takes the role [18], [19]: *A capacitied network G contains χ pair-wise capacity-disjoint unit spanning trees, if and only if for every partition that separates the network into k components, the total cross-component link capacity is at least $(k-1)\chi$.*

Unicast and broadcast are special cases of multicast, with the number of receivers being 1 and n, respectively, where $n = |V|$ is the size of the network. Consequently, Theorem 1 is a generalization of both the max-flow min-cut theorem and Tutte-Nash-Williams' theorem. For any given cut (vertex partition) of the network, we can assign a distance 1 to each link in the cut (partition), and a distance 0 to all the other links. Then the condition in Theorem 1 implies the cut condition (the partition connectivity condition) in the max-flow min-cut theorem (Tutte-Nash-Williams' theorem).

A bandwidth efficiency perspective

Since the total bandwidth capacity of a network is fixed, the achievable multicast rate closely depends on the bandwidth efficiency of the multicast transmission. Generally speaking, the higher the bandwidth efficiency, the higher the achievable multicast rate. Theorem 1 essentially claims that these two quantities are exactly proportional to each other, once we account for the fact that prolonging or shrinking an internal branch without changing its capacity does not affect the achievable multicast rate. We now reformulate Theorem 1 in this direction, after giving two definitions. A *link contraction* means replacing an 2-hop internal path u-z-v (*internal* means degree of z is 2) with a link uv, and set $C(uv) = \min\{C(uz), C(zv)\}$. *Link expansion* is the inverse operation for link contraction, where a link uv is replaced with a 2-hop path u-z-v, with $C(uz) = C(zv) = C(uv)$.

Theorem 1.a. For a multicast connection in an undirected network G, a sequence of link contraction and link expansion operations can be applied on G, after which the maximum multicast rate equals to the bandwidth capacity of the network divided by the minimum bandwidth consumption required for multicasting one bit information.

V. Efficient solution: the subgradient algorithm

In order to construct a subgradient solution for the maximum multicast rate problem, we have the choices of applying Lagrangian relaxation on either constraints in the primal program (dual subgradient), or constraints in the dual program (primal subgradient). We have decided to take the later approach, due to the following facts. First, dual subgradient methods do not always yield optimal primal solutions, which contain the optimal routing information we need. Second, as we will show, our primal subgradient algorithm decomposes the entire problem into a sequence of max-flow/min-cut computations, and allows appealing combinatorial interpretations. We now present the primal subgradient solution in three steps: the dualization strategy, subgradient iterations, and maximum rate computation.

A. The dualization strategy

Consider the dual linear program given in Sec.III-C for the maximum multicast rate problem. We choose to relax constraint group (5), which corresponds to primal variables $c(\overrightarrow{uv})$. Recall that $c(\overrightarrow{uv})$ specifies the capacity of each directed link, and therefore determines an orientation of the original undirected network. The objective function is modified to:

$$\begin{aligned}
& \textstyle\sum_{uv} C(uv)x(uv) + \sum_{\overrightarrow{uv}} c(\overrightarrow{uv})(\sum_i y_i(\overrightarrow{uv}) - x(uv)) \\
= \; & \textstyle\sum_{uv} x(uv)(C(uv) - c(\overrightarrow{uv}) - c(\overrightarrow{vu})) \\
& \textstyle + \sum_{\overrightarrow{uv}}(c(\overrightarrow{uv}) \sum_i y_i(\overrightarrow{uv})) \\
= \; & \textstyle\sum_i \sum_{\overrightarrow{uv}} c(\overrightarrow{uv}) y_i(\overrightarrow{uv}) - \sum_{uv} x(uv)\Delta(uv)
\end{aligned}$$

Note when $\Delta(uv) > 0$ for any uv, the modified objective function does not have a lower bound, with $x(uv)$ freely chosen from $[0, \infty)$. Therefore dual feasibility requires $\Delta \leq 0$, *i.e.*, $c(\overrightarrow{uv}) + c(\overrightarrow{vu}) \leq C(uv)$, $\forall uv$.

The Lagrangian dual we obtain is then:

Maximize $\quad L(c)$

Subject to:

$$\begin{cases} c(\overrightarrow{uv}) + c(\overrightarrow{vu}) \leq C(uv) & \forall uv \\ c(\overrightarrow{uv}) \geq 0 & \forall\, \overrightarrow{uv} \end{cases}$$

where

$$L(c) = \text{Min}_{P_2} \sum_i \sum_{\overrightarrow{uv}} c(\overrightarrow{uv}) y_i(\overrightarrow{uv}) \qquad (5.1)$$

with P_2 being the polytope:

$$P_2: \begin{cases} y_i(\overrightarrow{uv}) + p_i(v) \geq p_i(u) & \forall i, \forall\, \overrightarrow{uv} \neq \overrightarrow{T_i S} \\ p_i(T_i) - p_i(S) \geq z_i & \forall i \\ \sum_i z_i \geq 1 & \\ y_i(\overrightarrow{uv}), z_i \geq 0 & \forall i, \forall\, \overrightarrow{uv} \end{cases}$$

Two critical observations justify our choice of the dualization strategy above. First, the price variables introduced through relaxation and optimized through subgradient iteration, c, is exactly the orientation of the network, the optimal values of which is essential to decide the maximum multicast rate and the optimal routing strategy. Second, the minimization subproblem (5.1) is separable, and may be decomposed into k min-cut computations. We shall come back to these two facts in the presentation of the subgradient iterations and the maximum rate computation, respectively.

B. Subgradient iterations

Choosing the initial primal solution

To start the subgradient iterations, we need a valid set of initial values for $c(\overrightarrow{uv})$, *i.e.*, an initial orientation of the multicast network. A possible choice that is promising both in theory and in practice, is to set $c[0](\overrightarrow{uv}) = \frac{1}{2}C(uv)$, $\forall\, \overrightarrow{uv}$. Using Nash-Williams' graph orientation theorem (strong version) [20], it can be shown that such a balanced orientation is 2-competitive, *i.e.*, if the maximum multicast rate in an optimal orientation is χ^*, then the balanced orientation may support a rate of at least $\frac{1}{2}\chi^*$ [3].

Updating dual variables

During each round k, given current values of $c[k]$ we solve subproblem (5.1) to obtain new dual values in $y[k]$. As previously mentioned, this subproblem has a nice separable structure, in the form of a weighted minimum cut computation. Note that when $\sum_i z_i = 1$,

$$\begin{aligned}
L(c) &= \text{Min}_{P_2} \textstyle\sum_i \sum_{\overrightarrow{uv}} c(\overrightarrow{uv}) y_i(\overrightarrow{uv}) \\
&= \text{Min}_i [\text{Min}_{P_3} \textstyle\sum_{\overrightarrow{uv}} c(\overrightarrow{uv}) y_i(\overrightarrow{uv})]
\end{aligned}$$

where P_3 is the standard cut polytope:

$$P_3 : \begin{cases} y(\overrightarrow{uv}) + p(v) \geq p(u) & \forall\ \overrightarrow{uv} \neq \overrightarrow{T_i S} \\ p(T_i) - p(S) \geq 1 & \\ y(\overrightarrow{uv}) \geq 0 & \forall\ \overrightarrow{uv} \end{cases}$$

i.e., the weighted minimum cut equals to the minimum cut when all weights sum to 1. Further note that $\sum_i z_i = 1$ must be satisfied in any optimal solution, since dual complementary slackness conditions require $\chi(\sum_i z_i - 1) = 0$. Therefore, for our specific problem, we can compute $y[k]$ by first computing k minimum cuts, *i.e.*, one minimum cut between the sender S and each receiver T_i:

$$y_i^* = \text{argmin}_{y \in P_3} \sum_{\overrightarrow{uv}} c[k](\overrightarrow{uv}) y(\overrightarrow{uv})$$

Then let $j = \text{argmin}_i \sum_{\overrightarrow{uv}} c[k](\overrightarrow{uv}) y_i^*(\overrightarrow{uv})$, we update y as follows:

$$y_j[k] = y_j^*, \textit{and}$$

$$y_i[k] = 0, \forall i \neq j.$$

Updating primal variables

Primal variables in the orientation c are updated in two steps. First, we compute a new orientation vector c' as follows:

$$c' = c[k] + \theta[k] \sum_i y_i[k] \qquad (5.2)$$

where θ is a prescribed sequence of step sizes. The new vector c' is not feasible in general. Therefore we need to project it into the feasible simplex, to obtain a valid new vector for updating c. One possible way of projection is to take a feasible point that is nearest to c':

$$c[k+1] = \text{argmin}_{c \geq 0, \Delta \leq 0} ||c - c'|| \qquad (5.3)$$

Here $||l||$ denotes the geometrical length of a vector l, *i.e.*, for $l = (l_1, \ldots, l_h)$, $||l|| = (\sum_{i=1}^h l_i^2)^{1/2}$. Another simpler way of projection, is to normalize c' according to:

$$c[k+1](\overrightarrow{uv}) = \begin{cases} c'(\overrightarrow{uv}) & \Delta'(uv) \leq 0 \\ \frac{c'(\overrightarrow{uv})}{c'(\overrightarrow{uv}) + c'(\overrightarrow{vu})} C(uv) & \Delta'(uv) > 0 \end{cases} \qquad (5.4)$$

where $\Delta'(uv) = c'(\overrightarrow{uv}) + c'(\overrightarrow{vu}) - C(uv)$. After both primal and dual variables are updated, the next iteration starts.

Step size selection and convergence

Step size rules play an important role in subgradient optimization. It governs both the ultimate convergence in theory, and the speed of convergence to optimal solution in practice. Large step sizes may be unstable, while small step sizes lead to slow convergence speed. Therefore it is common practice to use varying step sizes: take a small number of large steps to reach the proximity of the optimal solution, then switch to small steps to avoid overhitting. In our case, where the original program is linear, designing step sizes that satisfy the following conditions will guarantee convergence:

$$\theta[k] \geq 0, \lim_{k \to \infty} \theta[k] = 0, \text{and} \sum_{k=1}^{\infty} \theta[k] = \infty$$

One simple sequence that satisfies the conditions above, is $\theta[k] = a/(bk + c)$, for some positive constants a, b and c. Below we give an example to illustrate the input, output, and convergence of the proposed algorithm.

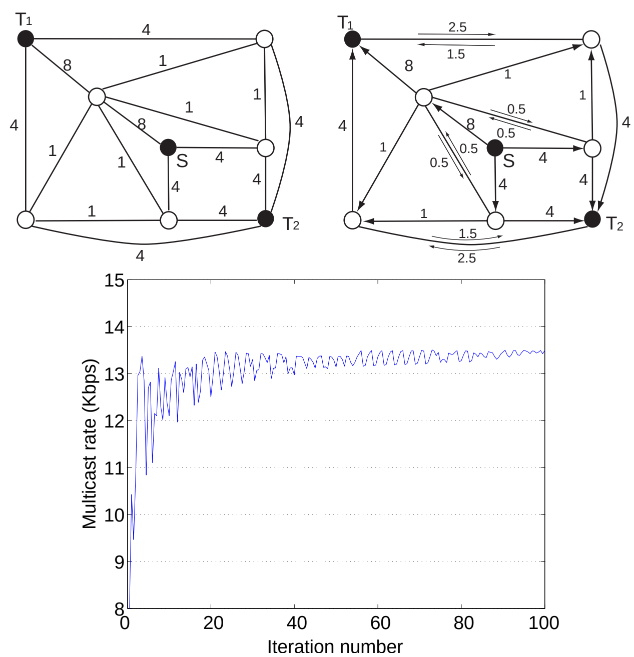

Fig. 2. A test case of the subgradient algorithm: input network, output orientation, and convergence sequence.

In the example shown in Fig. 2, S is the multicast sender, T_1 and T_2 are the multicast receivers. The maximum multicast rate possible is 13.5. Rate computed by the subgradient algorithm converges to range $[13.4, 13.5]$ within 100 iterations. The network in this example is actually among the most adversary to our algorithm, in that network flows towards different receivers constantly compete for link bandwidth in opposite directions. Our experiences show that the convergence speed is usually much faster for randomly generated multicast networks.

Algorithm interpretation

We now take a retrospect at the subgradient algorithm just presented, and show that it has a very appealing combinatorial interpretation. First, the algorithm takes a guessed orientation of the network as a starting point. Then during each iteration, it updates the orientation according to (5.2), (5.3) and (5.4). In (5.2), larger values for $\sum_i y_i[k]$ leads to larger values for c', which in turn leads to larger values for $c[k+1]$ in (5.3)(5.4). Note that non-zero values for $y_i(\overrightarrow{uv})$ means the link uv is in the S-T_i min-cut, and is therefore the "bottleneck" for the $S \to T_i$ transmission. From the flow perspective, non-zero values of $y(\overrightarrow{uv})$ means $f(\overrightarrow{uv}) = c(\overrightarrow{uv})$, since dual complementary slackness conditions require $y_i(\overrightarrow{uv})(f(\overrightarrow{uv}) - c(\overrightarrow{uv})) = 0$. Therefore links with non-zero y_i values are saturated links in the $S \to T_i$ max-flow. We conclude that the new capacity allocation in (5.2), (5.3) and (5.4) favors links with larger $\sum_i y_i[k]$ values, which are links that are more saturated.

Therefore, during each iteration of orientation refinement, the algorithm computes the max-flow/min-cut from the sender to each receiver, and increases the capacity share for more saturated links, while decreases the capacity share for under-utilized links. This has been summarized in Table I.

TABLE I

MAXIMUM MULTICAST RATE: SOLUTION SUMMARY

(1) *Choose initial orientation (*e.g., *balanced orientation)*

(2) Repeat
  Compute $S \to T_i$ max-flow, $\forall i$
  Refine orientation:
    increase bandwidth share for saturated links
    decrease bandwidth share for under-utilized links
 Until convergence
$\to$ optimal orientation obtained

(3) *Compute $S \to T_i$ max-flow, $\forall i$*
$\to$ optimal multicast rate and routing strategy obtained

(4) *Randomized code assignment*
$\to$ complete transmission strategy obtained

C. Computing the maximum rate

When the subgradient algorithm converges, it yields optimal primal values in c, but not necessarily optimal dual values in y — the dual values upon convergence may not even be feasible. Although there exist convex combination techniques to recover these optimal dual values [9], [21], it is not necessary in our solution. We can directly recover the whole set of optimal primal values from optimal values in c.

Recall that a feasible vector c specifies an orientation of the undirected network. Therefore optimal values of c give an optimal orientation. Once the orientation is determined, the undirected maximum multicast rate problem boils down to a directed one, *i.e.*, computing the maximum multicast rate in a directed network. By the result on directed multicast rate feasibility proven by Ahlswede *et al.* and Koetter *et al.*, this can be accomplished by invoking a maximum flow computation from sender S to each of the k receivers T_i. Let f_i^* denote the resulting $S \to T_i$ flow vector, and $|f_i^*|$ denote the corresponding flow rate. Then our final solution to the maximum multicast rate problem is:

- maximum multicast rate: $\chi = \min_i |f_i|$
- optimal routing strategy of information flows: f^*, where $f^*(\overrightarrow{uv}) = \max_i f_i(\overrightarrow{uv})$, $\forall\ \overrightarrow{uv} \in A$

As an illustration, the two network flows computed in the previous example are shown in Fig. 3.

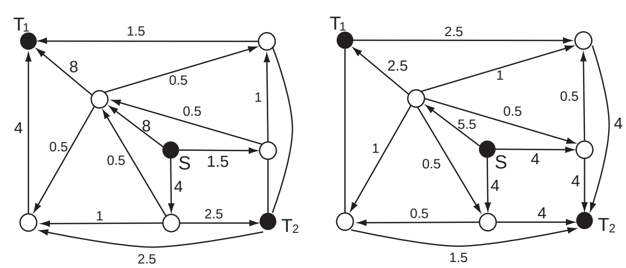

Fig. 3. Output network flow to each multicast receiver.

D. Discussions on distributed implementation

Beside simplicity and efficiency, the potential for distributed implementation remained as another goal during our design of the subgradient algorithm. After all, all protocols that work in real-world networks need to be decentralized. we now take a step-by-step examination of the entire solution, and discuss how each step can be transferred into distributed, pure local computations, where each node maintains only local information about its incident links and one-hop neighbors.

In the initialization phase of the dual subgradient algorithm, it is sufficient to have each node u compute its local orientation, by setting $c(\overrightarrow{uv}) = c(\overrightarrow{vu}) = \frac{1}{2}C(uv)$, for each of its incident link uv.

Primal variable update is achieved through pure local computation, since each node can update the capacity of an incident directed link $\overrightarrow{uv}$ according to (5.2), based on current values of local variables $c[k](\overrightarrow{uv})$, $\theta[k]$ and $y[k](\overrightarrow{uv})$.

Most computation in the subgradient algorithm is performed in dual variable updates, and in the final maximum flow rate computation. Each of these steps translates into k max-flow/min-cut computations. As previously mentioned, various efficient algorithms exist for the classical max-flow/min-cut problem, some of which permits natural distributed implementations, such as the push-relabel algorithm [5] and the ϵ-relaxation algorithm [8]. For example, throughout the execution of the distributed version of the push-relabel algorithm, each node exchanges messages with its direct neighbors only, and maintains information about capacities and flow rates on its incident links, plus distance labels of its neighbors and its own.

So far we have shown that our algorithm for computing the optimal multicast routing strategy can be implemented in a distributed fashion. In order to utilize such optimal routing strategy in data transmission, we need to further decide how each node linearly combines its incoming information flows to form its outgoing information flows. A simple distributed solution to this code assignment problem is randomized coding [22], in which each node just locally generates a random code matrix, without any message-passing required at all. With mild assumptions on the size of the base field for coding operations, the chance of generating a conflict is negligibly small [22].

VI. Conclusion

The main problem of interest in this paper is to achieve the maximum multicast transmission rate in an undirected network. We first formulate the problem as linear network optimization. We then apply Lagrangian relaxation on the primal problem, and derive a necessary and sufficient condition for multicast rate feasibility. Our condition is a generalization of the well-known conditions for the unicast case and the broadcast case. We next construct a subgradient algorithm that solves the undirected version of the maximum multicast rate problem in an efficient and distributed manner, by decomposing the problem into a sequence of max-flow/min-cut computations. Combined with randomized code assignment, which incurs essentially zero overhead in both computation and communication, our algorithm constitutes a promising approach for generating the entire maximum-rate multicast strategy.

Acknowledgments

The authors would like to thank Jun Yuan for helpful discussions on the material presented in Section V, and the anonymous reviewers for their valuable feedback that helped revising the paper.

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] R. Koetter and M. Médard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

[3] Z. Li and B. Li, "Network Coding in Undirected Networks," in *Proc. of the 38th Annual Conference on Information Sciences and Systems (CISS)*, 2004.

[4] Z. Li, B. Li, D. Jiang, and L. C. Lau, "On Achieving Optimal Throughput with Network Coding," in *Proceedings of IEEE INFOCOM*, 2005.

[5] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, New Jersey, 1993.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. of ACM SIGCOMM*, 1994, pp. 212–225.

[7] LAN MAN Standards Committee of the IEEE Computer Society, "IEEE 802.11: Wireless LAN MAC and PHY Specifications, Chapter 11," 1999.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, 1989.

[9] D. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving Minimum-Cost Multicast: A Decentralized Approach Based on Network Coding," in *Proceedings of IEEE INFOCOM*, 2005.

[10] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371, 2003.

[11] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial Time Algorithm for Network Information Flow," in *Proceedings of the 15th ACM Symposium on Parallelism in Algorithms and Architectures*, 2003.

[12] L. C. Lau, "An approximate max-Steiner-tree-packing min-Steiner-cut theorem," in *Proc. of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[13] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner Trees," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[14] Y. Zhu, J. Guo, and B. Li, "Multicast with Network Coding in Application-Layer Overlay Networks," *Journal on Selected Areas in Communications (JSAC), Special Issue on Service Overlay Networks*, vol. 22, no. 1, 2004.

[15] P. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. of the 51st Annual Allerton Conference on Communication, Control and Computing (Allerton)*, 2003.

[16] M. Wang, Z. Li, and B. Li, "A Case for Coded Overlay Flows," Tech. Rep., ECE, University of Toronto, July 2004.

[17] *GNU Linear Programming Kit*, http://www.gnu.org/ software/glpk/glpk.html.

[18] W. T. Tutte, "On the Problem of Decomposing a Graph Into n Connected Factors," *J. London Math. Soc.*, vol. 36, pp. 221–230, 1961.

[19] C. St. J. A. Nash-Williams, "Edge-disjoint Spanning Trees of Finite Graphs," *J. London Math. Soc.*, vol. 36, pp. 445–450, 1961.

[20] J. Bang-Jensen and G. Gutin, *Digraphs:Theory, Algorithms and Applications*, Springer Verlag, 2002.

[21] H. D. Sherali and G. Choi, "Recovery of Primal Solutions When Using Subgradient Optimization Methods to Solve Lagrangian Duals of Linear Programs," *Operations Research Letters*, vol. 19, 1996.

[22] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," in *41st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2003.