Multirate Media Streaming Using Network Coding

Niveditha Sundaram Parameswaran Ramanathan Suman Banerjee

Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706–1691
nsundara, parmesh @ece.wisc.edu, suman@cs.wisc.edu

Abstract

Multimedia data transfers typically involve large volumes of data. Multirate multicast transmissions using layered source coding are generally used to deliver data streams to heterogeneous receivers. Network coding has been envisioned to increase throughput and deliver higher data rates than conventional source coding or no coding. The paper proposes a polynomial time algorithm for multicast to heterogeneous receivers using network coding. The overall goal of the algorithm is to maximize the aggregate rate to all the receivers. The problem is formulated as a linear programming optimization and solution from this optimization is used to assign linear network codes to all nodes using the Linear Information Flow (LIF) algorithm described in [1]. Empirical evaluation of the proposed solution shows that all receivers can be given a rate equal to their max-flows in all of the simulated instances.

I. Introduction

With the emergence of multimedia applications in business and entertainment, demand for real-time multi-point applications such as multi-party gaming, video-conferencing and video-on-demand services have increased. Multimedia data transfers typically contain large volumes of data, and hence redundant unicast transmission of the same data to multiple receivers is likely to consume excessive network resources. Conventional unirate multicasting is a solution to reduce the resource consumption. However, if the receivers in same multicast session differ in their max-flow rate from the source, unirate multicasting either overwhelms slow receivers or starves the fast ones. Hence, multirate multicasting, where individual receiver rates depend on their max-flow rates, is a preferred mode for distributing large content applications.

One approach to achieve multirate multicasting is layered coding [2–6]. In layered coding, the source encodes the data stream into a base layer and several enhancement layers. Receivers subscribe to a layer cumulatively, i.e, if a receiver subscribes to layer k, it also receives layers 1 $\ldots k$. The layers are incrementally combined at the receiver to provide progressive refinement. Rate control algorithms for multirate multicasting strive to make efficient use of the network resources [2–4, 7]. The objective of the algorithms is to deliver a rate to each receiver that is dependent only on its flow rate from the source. Issues such as max-min fair rate allocation and utility-fair rate allocation have been addressed in [6, 8–12]. These schemes use layered coding to achieve the desired fairness.

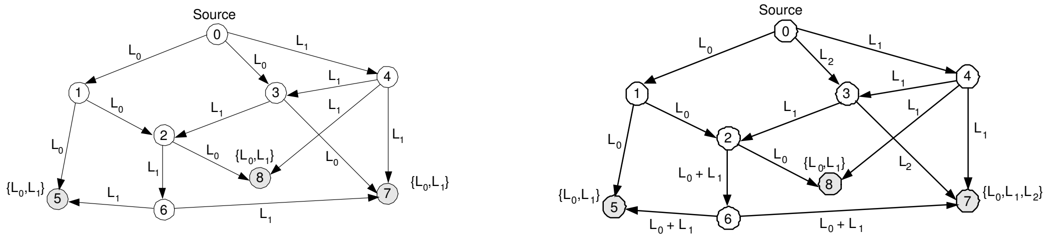
The scheme proposed in this paper also uses layered coding for multirate multicasting. However, the proposed scheme also exploits a recently proposed idea called *network coding* [1, 13–16]. In network coding, the source and the intermediate nodes have the additional capability of encoding streams. It has been envisioned as a means to reduce delay, increase throughput, and improve reliability of a given network. Literature in network coding can be classified into two distinct categories, one focusing on the information theoretic aspects and the other on applications. On the information theoretic side, [13, 14] proved that, using network coding, each receiver can be given a rate equal to the minimum of the maximum flow from the source to each of the receivers. An algebraic framework identifying the networks code for each node to achieve the desired rates in presented in [15]. Polynomial time algorithms to identify linear network codes for all nodes to deliver minimum of the maximum flow rate to all receivers is proposed in [1, 16]. Implementation-oriented schemes for multicast media streaming in overlay networks and for file distribution to large number of cooperative users are respectively proposed in [17] and[18].

To the best of our knowledge, all of the work in literature on network coding except [15] consider homogeneous receivers, i.e., all receivers are given the same data rate irrespective of the available bandwidth. In [15], heterogeneous receivers are considered and algebraic conditions are derived for existence of solutions given different data needs of the receivers. However, an efficient algorithm to find the network codes that simultaneously meet the given data needs of the receivers are not presented in [15]. In contrast, this paper presents a polynomial time algorithm for finding the network codes for heterogeneous receivers in the context multimedia streaming with layered coding, i.e., the heterogeneity among the receivers is in the number of layers (consequently, the data rates) delivered to them. The overall goal is to deliver as many layers as possible to each receiver. This overall goal is formulated as maximizing the sum of the number of layers delivered over all receivers. The key aspect of the paper is that it describes a polynomial time algorithm for identifying the network codes to be used by each node to meet the overall goal. Empirical evaluation of the proposed algorithm are also included.

The rest of the paper is organized as follows. The model and the problem statement are presented in Section II. A brief review of the scheme from [1] is presented in Section III. The proposed solution is described in Section IV. An empirical evaluation of the solution is included in Section V. The paper concludes in Section VI.

II. Model and Problem Statement

The network is modeled as a weighted directed graph $G = (V, E)$, where V is the set of nodes and E is the set of weighted directed edges between them. The weight on a edge $(u, v) \in E$ represents the bandwidth available between nodes u and v. One of the nodes in V is the source node that generates a multimedia stream and a subset $R \subseteq V$ is a set of receivers for the multimedia stream. The stream is encoded as a set of layers L_0, L_1 …, L_N such that to decode a layer L_i a receiver needs all layers L_1, …, L_i. Furthermore, the layers are such



(a) Digraph without Network Coding

(b) Digraph with Network Coding

Fig. 1. Illustration of network coding.

that the quality of decoded stream depends on the number of layers a receiver can decode; the more the layers, the higher the quality.

Each node applies linear network codes on the data stream it receives before forwarding them on the output edges. Specifically, consider a typical node $u \in V$. Let P_u denote the set of nodes from which node u directly receives a data stream, i.e., $P_u = \{v : (v,u) \in E$ and v forwards to $u\}$. Similarly, let S_u denote the set of nodes to which node u forwards a data stream, i.e., $S_u = \{v : (u,v) \in E$ and u forwards to $v\}$. Further, let $d_{(v,u)}$ denote the data stream node u receives on input edge (v,u). Then, the data stream forwarded by node u on an output edge (u,v) is

$$d_{(u,v)} = \sum_{w \in P_u} \alpha_{(w,u),(u,v)} * d_{(w,u)}, \quad \text{for each } v \in S_u,$$

where $\alpha_{(w,u),(u,v)}$ is a coefficient taken from a Galois Field of suitable dimension. The vector of α's corresponding to an edge (u,v) is referred to as the network code applied by node u for edge (u,v).

Because of network coding, a node $i \in R$ may not directly receive L_0, $\ldots L_N$. Instead, it may receive a linear combination of the layers. If node $i \in R$ receives an appropriate set of linearly combined layers, then it can decode them to construct a subset of the layers L_0, $\ldots L_N$. The receiver can further decode the layers to reconstruct the multimedia stream. More specifically, consider a receiver $i \in R$. Suppose that it receives data $d_{(v,i)}$ from each $v \in P_i$. If the network codes are properly selected at all nodes, then the receiver i can decode the layers L_0, $\ldots$, $L_{|P_i|}$ as follows. The receiver i first selects $|P_i|^2$ coefficients from the Galois Field, denoted as $\beta_{j,(v,i)}$, $1 \leq j \leq |P_i|$ and $v \in P_i$. Then, it decodes

$$L_j(i) = \sum_{v \in P_i} \beta_{j,(v,i)} * d_{(v,i)} \quad \text{for each } 1 \leq j \leq |P_i|.$$

For example, consider the network shown in Figure 1(a) and 1(b). Assume all links are of unit capacity. Node 0 is the source node and nodes 5, 7, and 8 are the receivers. It can be shown that, without network coding, receivers 5, 7, and 8 can be given at most two layers L_0 and L_1. One such solution without network coding is shown in Figure 1(a). However, with network coding, receiver 8 can be given an additional layer L_2. Figure 1(b) shows a solution with network coding that deliver layers L_0 and L_1 to receivers 5 and 8 and L_0, L_1, and L_2 to receiver 7. This is possible because node 2 linearly combines the two data streams it receives from nodes 1 and 3 before “forwarding“ both streams on the unit capacity link between nodes 2 and 6. As a result, receivers 5 and 8 receive L_0 and $L_0 + L_1$, from which they reconstruct L_0 and L_1. Receiver 7, on the other hand, directly receives L_0, L_1 and L_2. Thus, network coding increases the aggregate data rate that can be delivered to the receivers.

Note that, in general, a node may not receive data on all its input edges and likewise it may not forward data on all its output edges. The total date rate delivered to receiver i is the number of layers it can decode. Let r_i denote the rate assured to receiver $i \in R$. Further, let m_i denote the maximum flow rate from source to i in the network. The paper assumes that the r_i must satisfy the following

Weak Fairness Criterion: $m_i \leq m_j \Rightarrow r_i \leq r_j$,

i.e., a receiver with larger maximum flow is given a higher data rate.

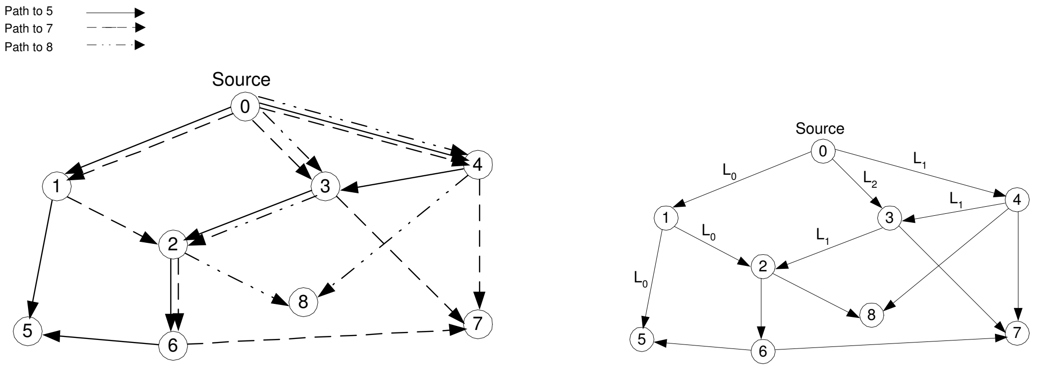
The problem then is to find the network codes for all nodes to maximize the total rates to all receivers while satisfying the weak fairness criterion.

III. Review of LIF Algorithm

The solution proposed in this paper builds on the LIF algorithm in [1]. Therefore, we briefly review the LIF algorithm. The LIF algorithm works as follows. Suppose s is the source node and R is the set of receivers. Let m_i denote the max-flow from s to receiver $i \in R$. For simplicity of presentation, assume all links in the network have unit capacity.

The LIF algorithm begins by constructing m_i edge disjoint paths from the s to receiver i, for each $i \in R$. It then steps through the nodes in topological order. In each step, it assigns a network code to the outgoing edges from the node. Since the nodes are considered in topological order, when a node u is being considered, all its incoming edges would have already been assigned a network code. The network code to an outgoing edge is assigned based on the following observation: if the data stream assigned to the edges in any min-cut from the source to a receiver are linearly independent, then the receiver can reconstruct the multimedia stream transmitted by the source. More specifically, consider a receiver $i \in R$. Let C_i be any set of m_i edges such that it contain exactly one edge from each m_i edge-disjoint paths from s to i. C_i is min-cut from s to i. Therefore, if the data stream on the edges in C_i are linearly independent, then receiver i can reconstruct the multimedia stream. Note that, if a edge belongs to a path for more than one receiver, then linear independence property must be satisfied with respect to each receiver.

For example, consider the network in Figure 1. As stated earlier, the max-flows to receivers 5, 7, and 8 are 2, 2, and 3, respectively. Figure 2(a) shows the 2, 3, and 2 paths to the receivers 5, 7, and 8 (indicated by solid lines for 5, dashed lines for 7, dash-dot lines for 8). Suppose that, in stepping through the nodes, the LIF algorithm has assigned the data stream to the outgoing edges of nodes 1, 3, and 4 as shown in Figure 2(b) and it is currently considering the edge (2, 6) of node 2. The edge (2,6) belongs on the paths of receivers 5 and 7. The set



(a) Paths to receivers $5, 7$ and 8 in digraph Figure 1.

(b) Example network code for edge $(2, 6)$.

Fig. 2. LIF code allocation example.

$\{(0, 1), (2, 6)\}$ is a min-cut based on the paths to receiver 5 and $\{(0, 3), (0, 4), \text{and}(2, 6)\}$ is a min-cut based on the paths to receiver 7. If the edge (2, 6) is assigned a network code $L_0 + L_1$, then the data on the both of the above min-cuts will be linearly independent. In [1], the authors prove that one can always find such a linearly independent code in polynomial time (if all the receivers are homogeneous).

IV. Proposed Solution

The proposed solution approach is comprised of the following six steps.

Algorithm Heterogeneous Receiver Rate_Estimate

1) Partition the set of receivers R into subsets $R_i \ldots R_k$ such that: (i) the receivers in each subset R_i, $1 \leq i \leq k$ have identical max flows, and (ii) the max flow of receivers in R_i is greater than that of receivers in any R_j, if $j < i$.
2) Construct k graphs $G_1 \ldots G_k$ as follows $G_j \equiv (V_j, E_j)$, where $V_j = V - \bigcup_{l=1}^{j-1} R_l$ and $E_j = \{(u, v) \in E : u, v \in V_j\}$.
3) Construct and solve a linear programming problem (see Figure 3) to determine r_j, for all $1 \leq j \leq k$, which are the rates delivered to receivers in R_j and parameters $Y_{(u,v)j}$ where $(u, v) \in E_j$ for each $1 \leq j \leq k$.
4) Augment the graphs $G_1 \ldots G_k$ with edge weights as determined from the linear programming problem. In particular, edge (u, v) in graph G_j is assigned a weight $Y_{(u,v)j}$.
5) Using Ford-Fulkerson max flow algorithm, determine paths to receivers in G_j, for all $1 \leq j \leq k$ such that each receiver gets a flow of $r_j - r_{j-1}$.
6) Use LIF algorithm [1] on the paths identified in Step 5 to deliver net rate r_j to each receiver in R_j for all $1 \leq j \leq k$.

Figure 3 shows the linear program formulation. The variables in the linear programming problem have the following semantics.

Linear Programming Problem /* Executed by source */

Maximize $\sum_{i=1}^{R} r_i$ subject to

1. $\sum_{(0,v)\in E_j} f_{i(0,v)j} = r_j - r_{j-1} \qquad \forall i \in (R \cap V_j), \forall 1 \leq j \leq k$

2. $\sum_{(u,i)\in E_j} f_{i(u,i)j} = r_j - r_{j-1} \qquad \forall i \in (R \cap V_j), \forall 1 \leq j \leq k$

3. $\sum_{u\in I_j(v)} f_{i(u,v)j} = \sum_{u\in O_j(v)} f_{i(v,u)j} \qquad \forall v \in V_j \forall 1 \leq j \leq k$
   $I_j(v) = \{u : (u,v) \in E_j\}$
   $O_j(v) = \{u : (v,u) \in E_j\}$

4. $Y_{(u,v)j} \geq f_{i(u,v)j} \qquad \forall i \in (R \cap V_j)$, where $1 \leq j \leq k$

5. $\sum_{j=1}^{q_{(u,v)}} Y_{(u,v)j} \leq C_{(u,v)} \qquad \forall (u,v) \in V$, where $q_{(u,v)} = max\{l : (u,v) \in E_l\}$

6. $r_1 \geq m$ where m is the minimum max flow among receivers in G.

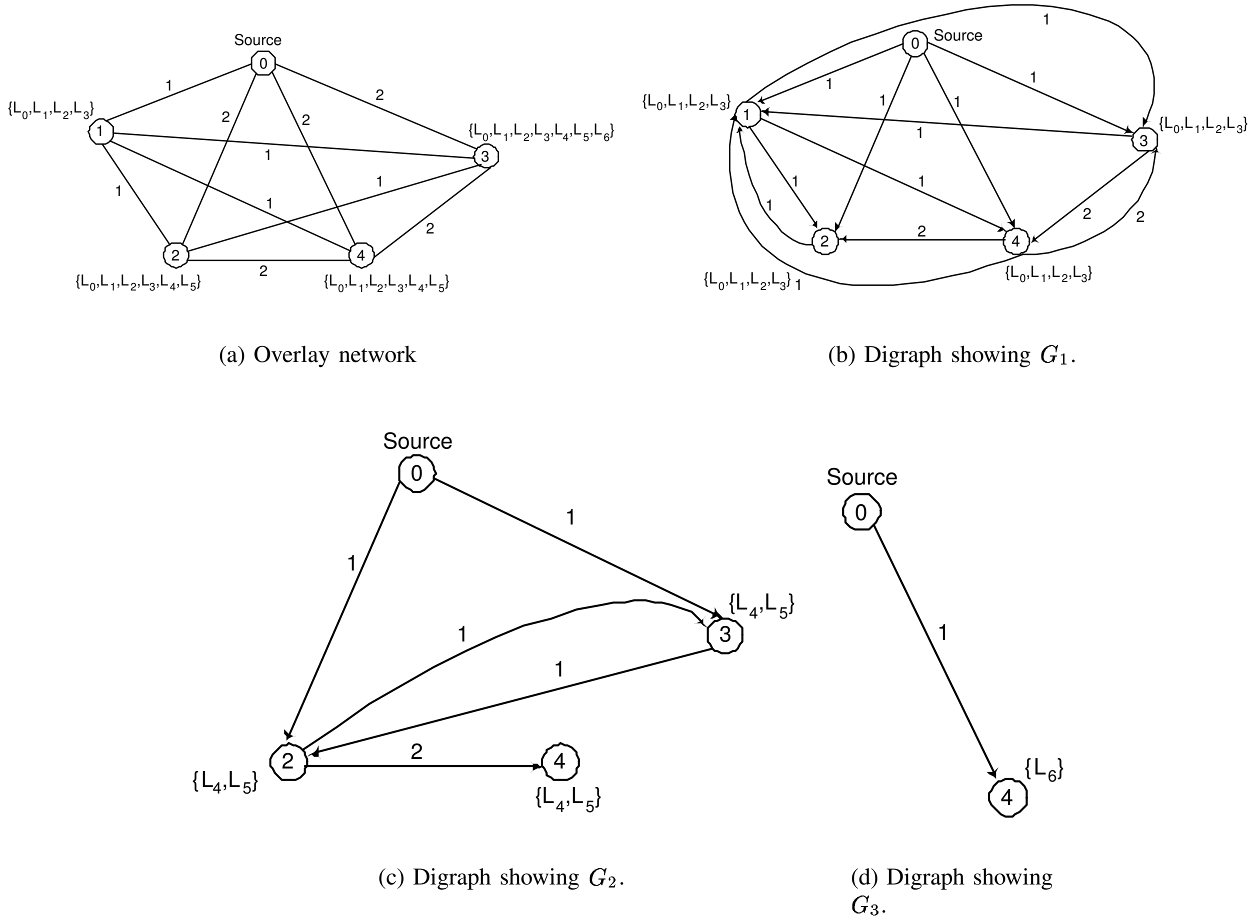
Fig. 3. Linear Program for maximizing the rates.

- $f_{i(u,v)j}$: Rate allocated to receiver i in subgraph G_j on edge (u,v).
- $Y_{(u,v)j}$: Total capacity allocated to all receivers in subgraph G_j on edge (u,v).

The constraints listed under $1, 2, 3$ in Figure 3 are flow balance constraints. Specifically, constraint 1 corresponds to flow balance constraints at the source. For all $1 \leq j \leq k$, flow balance constraints for the source include all outgoing edges $(0,v) \in E_j$. For $j = 1$, the total outgoing flow rate from the source is equal to r_1, which is the minimum max flow rate guaranteed to all receivers. r_0 is assumed to be 0. Constraint 2 corresponds to flow balance constraints at the receivers. Receiver flow balance constraints include all incoming edges $(u,i) \in E_j$ for $1 \leq j \leq k$. Similarly, constraint 3 gives the flow balance constraints for all intermediate nodes in the graph. For each node, the total outgoing flow rate is equal to the total incoming flow rate on each path to a receiver $i \in (R \cap V_j)$.

Constraints 4, 5 in Figure 3 are the network coding constraints. Constraint 5 gives the capacity constraint for each edge $(u,v) \in V$. The sum of all the $Y_{(u,v)j}$ for each (u,v) for all $1 \leq j \leq k$ can not exceed the capacity of edge (u,v). Finally, constraint 6 gives the minimum rate that should be allocated to all receivers, which is equal to the minimum max flow among all the receivers in G.

Example: Consider the overlay network shown in Figure 4(a) with link capacities as shown. Node 0 is the source and nodes $1, 2, 3, 4$ are receivers. The max flow to receivers $1, 2, 3, 4$ are $4, 6, 6, 7$ respectively, and therefore at the end of Step 1 in Algorithm Heterogeneous Receiver Rate_Estimate (HRRE) $R_1 = \{1\}$, $R_2 = \{2, 3\}$, and $R_3 = \{4\}$. Figure 4(b), 4(c) and 4(d) show the three subgraphs G_1, G_2 and G_3 that are obtained at the end of Steps $2, 3$ and 4 with the edge weights $Y_{(u,v)j}$ as shown, for $1 \leq j \leq 3$. Solving the linear program from Step 3 the rates are determined to be 4, 6 and 7 for r_1, r_2 and r_3 respectively. From Steps 5 and 6, the



(a) Overlay network

(b) Digraph showing G_1.

(c) Digraph showing G_2.

(d) Digraph showing G_3.

Fig. 4. Overlay network and the subgraphs G_1, G_2, and G_3.

network codes found for G_1, G_2 and G_3 are shown in Figure 5(a), 5(b) and 5(c). From the figures, it can be seen that the network codes deliver the promised rates of 4, 6, 6, and 7 to receivers 1, 2, 3, and 4, respectively. For instance, receivers 1, 2, 3, 4 get a rate of 4 from subgraph G_1, receivers 2, 3, 4 get 2 more from G_2, making a total of 6, while receiver 4 gets a rate of 1 more from G_3 achieving a total rate of 7 for receiver 4.

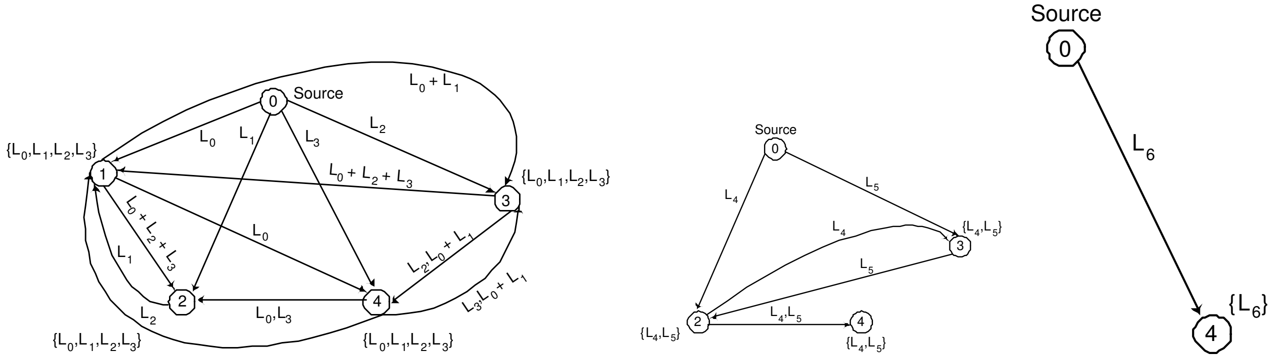
A. Proof of Correctness

The following theorem can be proved about Algorithm HRRE.

Theorem 1: Algorithm HRRE finds a set of network codes such that a receiver $j \in R_j$ gets rate r_j for all $1 \leq j \leq k$.

Proof:

The proof follows from the following observations:



(a) Digraph showing network coding using G_1.

(b) Digraph showing network coding using G_2.

(c) Digraph showing network coding using G_3.

Fig. 5. Network coding in subgraphs G_1, G_2, and G_3.

- The linear programming problem in Figure 3 always has a feasible solution because a solution that guarantees minimum of max flow to all receivers using LIF algorithm satisfies all constraints of the linear program.
- The minimum of max flow to all receivers in $G_j = r_j - r_{j-1} \forall 1 \leq j \leq k$. This follows from the flow balance constraints Steps 1, 2, 3 of the linear programming problem.
- All receivers in $R \cap V_j$ get a rate of $r_j - r_{j-1}$ in subgraph G_j using network coding $\forall 1 \leq j \leq k$. Follows from the above observation and the results in [1].
- Receivers in R_j get a rate of r_j. The receivers in R_j are in graphs G_1, $G_2 \ldots G_j$. Each receiver receives r_1 from G_1, $r_2 - r_1$ from G_2 and so on until $r_j - r_{j-1}$ from G_j.

Thus, all the receivers in R_j receive a rate of r_j for all $1 \leq j \leq k$. ■

V. Evaluation

This section contains results from an empirical evaluation of Algorithm HRRE (Heterogeneous Receiver Rate Estimate) described in Section IV. The objectives of the evaluation are as follows:

- demonstrate the advantage of using multirate multicast over unirate multicast for streaming
- quantify the benefit of using network coding over not using any coding
- compare the performance of HRRE for different topologies

The simulation environment is described as follows. The networks are based on an overlay architecture with ten fully-connected nodes with link capacities that uniformly distributed between 1 and 15. One of the nodes is the source of the multimedia stream and the other nine nodes are receivers.

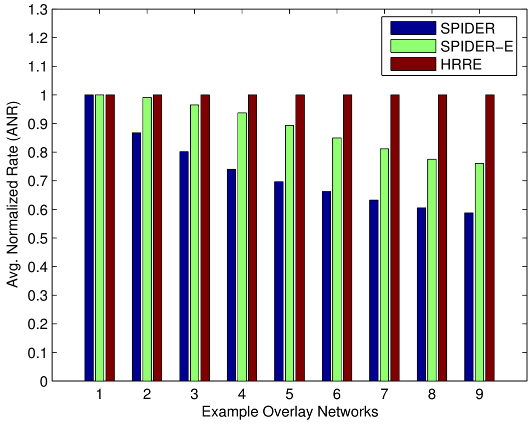


Fig. 6. ANR for different receivers for 10-node overlay network.

Figure 6 compares the performance of three algorithms for multimedia streaming, namely HRRE with network coding, unirate multicasting called SPIDER [19], and an enhanced version of SPIDER called SPIDER-E for multirate multicast without network coding. The comparison is with respect to a performance measure called Average Normalized Rate (ANR), where ANR is defined as follows. Let r_i be the rate delivered to receiver i in given scheme and let m_i be the max-flow rate from the source to receiver i. Then, ANR for that scheme is defined as $\frac{1}{|R|}\sum_{i \in R} \frac{r_i}{m_i}$. Clearly, larger the ANR, the better the scheme.

In Figure 6, the x-axis corresponds to the receivers $1 \ldots 9$ and the y-axis shows the ANR for each receiver for the three schemes. Note that the receivers are numbered in the increasing order of their max-flow rates, i.e., node 1 has the least max-flow rate and node 9 has the largest max-flow rate. For each node, the leftmost bar indicates basic SPIDER scheme for unirate multicast. [19] showed that SPIDER delivers the minimum of max flow rates to all receivers. Although SPIDER will not overwhelm a receiver with a lower max flow rate, it will starve the node with higher max-flow rate because it delivers the same rate to all receivers. The middle and the rightmost bar respectively show the results for SPIDER-E and HRRE. From the figure it can be seen that, for streaming data, multirate multicast with no coding (SPIDER-E) and with network coding (HRRE) deliver higher rates than unirate multicast (SPIDER). SPIDER-E shows an ANR increase of approximately 15% over SPIDER and HRRE shows an ANR increase of approximately 40% over SPIDER. It can be seen that for all the networks considered, HRRE achieves max flow rate to all receivers. Although SPIDER-E delivers the minimum max flow rate to receiver 1, and delivers rates that are higher than the minimum max flow rates to receivers close to 1, it starves all the higher rate receivers.

On the other hand, HRRE with network coding ensures that all receivers receive their respective max flow rates for all of the ten topologies that are considered. Since no receiver can receive at rates higher than its max-flow rate, HRRE achieved the best possible rates for all receivers in all the ten simulated networks.

VI. Conclusions

The paper presented an algorithm for finding network codes that maximize total rates to heterogeneous receivers. There are not any efficient algorithms in literature to find network codes to serve heterogeneous receivers. Algorithm Heterogeneous Receiver Rate_Estimate is the first polynomial time solution in literature that finds network codes to allocate rates to heterogeneous receivers. Empirical evaluations show that the algorithm achieves rates equal to the max flow rate to all receivers in all simulated topologies.

References

[1] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proceedings of SPAA 2003*, June 2003.

[2] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proceedings of ACM SIGCOMM 96*, Aug. 1996.

[3] X. Li, S. Paul, and M. Ammar, "Layered video multicast with retransmissions (lvmr): Evaluation of hierarchical rate control," in *Proceedings of IEEE INFOCOM 1998*, Mar. 1998.

[4] X. Li, S. Paul, and M. Ammar, "Multi-session rate control for layered video multicast," in *Proceedings of Multimedia Computing and Networking*, Jan. 1999.

[5] K. Nakauchi, H. Morikawa, and T. Aoyama, "A network-supported approach to layered multicast," in *Proceedings of ICC*, June 2001.

[6] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multirate multicast," in *Proceedings of IEEE INFOCOM 2001*, Apr. 2001.

[7] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus priority dropping for layered video," in *Proceedings of ACM SIGCOMM 98*, Sept. 1998.

[8] S. Sarkar and L. Tassiulas, "Fair allocation of utilities in multirate multicast networks," in *Proceedings of Allerton Conference on Communication, Control and Computing*, Sept. 1999.

[9] S. Sarkar and L. Tassiulas, "Distributed algorithms for computation of fair rates in multirate multicast," in *Proceedings of IEEE INFOCOM 2000*, Apr. 2000.

[10] K. Kar, S. Sarkar, and L. Tassiulas, "A scalable low-overhead rate control algorithm for multirate multicast sessions," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1541–1557, Apr. 2001.

[11] H-W. Lee, J-W. Cho, and S. Chong, "A distributed max-min flow control algorithm for multi-rate multicast flows," in *Proceedings of GLOBECOM 2004*, Dec. 2004.

[12] Y. Cui, Y. Xue, and K. Nahrstedt, "Max-min overlay multicast: Rate allocation and tree construction," in *Proceedings of IWQoS*, June 2004.

[13] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yueng, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[14] S-Y. R. Li, R. W. Yueng, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[15] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE Transactions on Networking*, October 2003.

[16] S. Jaggi, P. Chou, and K. Jain, "Low complexity algebraic multicast network codes," in *Proceedings of International Symposium on Information Theory*, July 2003.

[17] J. Guo, Y. Zhu, and B. Li, "Codedstream: Live media streaming with overlay coded multicast," in *Proceedings of SPIE*, Dec. 2003.

[18] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 1–13, Jan. 2004.

[19] S. Ganguly, A. Saxena, S. Bhatnagar, S. Banerjee, and R. Izmailov, "Fast replication in content distribution overlays," in *Proceedings of IEEE INFOCOM 2005*, Mar. 2005.