Optimal Distributed Broadcasting with Per-neighbor Queues in Acyclic Overlay Networks with Arbitrary Underlay Capacity Constraints

Shaoquan Zhang, Minghua Chen
The Chinese University of Hong Kong
{zsq008, minghua}@ie.cuhk.edu.hk

Zongpeng Li
University of Calgary
zongpeng@ucalgary.ca

Longbo Huang
Tsinghua University
longbohuang@mail.tsinghua.edu.cn

***Abstract*—Broadcasting system such as P2P streaming systems represent important network applications that support up to millions of online users. An efficient broadcasting mechanism is at the core of the system design. Despite substantial efforts on developing efficient broadcasting algorithms, the following important question remains open: How to achieve maximum broadcast rate in a distributed manner with each user maintaining information queues only for its direct neighbors? In this work, we first derive an innovative formulation of the problem over acyclic overlay networks with arbitrary underlay capacity constraints. Then, based on the formulation, we develop a distributed algorithm to achieve the maximum broadcast rate and every user only maintains one queue per-neighbor. Due to its lightweight nature, our algorithm scales very well with network size and remains robust against high system dynamics. Finally, by conducting simulations we validate the optimality of our algorithm under different network capacity models. Simulation results further indicate that the convergence time of our algorithm grows linearly with the network size, which suggests an interesting direction for future investigation.**

I. Introduction

Information broadcasting is an increasingly important application for content delivery and network control in the Internet. For example, in P2P streaming systems such as PPLive [1] and UUSee [2], the streaming server continuously generates streaming contents and disseminates them to all the participating users.

We study the optimal broadcasting problem, *i.e.*, maximizing the broadcast rate at which contents are received by all the users simultaneously. This fundamental problem is at the core of many broadcasting systems, attracting a substantial level of attention from both industry and academia. A number of work in the literature design broadcasting algorithms under various network models. We consider a general network model that subsumes most known ones such as the edge-capacitated network model and the node-capacitated network model, as illustrated in Fig. 1.

A classic work on broadcasting is due to Edmonds [3], who provided a centralized algorithm for computing the maximum broadcast rate. Edmonds' algorithm constructs a set of edge-disjoint spanning trees, each rooted at the source and reaching every user in the network. However it works only under the edge-capacitated network model. Under the node-capacitated network model, Sengupta *et al.* [4] formulated the problem as finding the set of spanning trees with maximum aggregate broadcast rate, subject to node capacity constraints. They then solved this optimization problem approximately by adapting the Garg-Konemann technique [5]. These two *centralized* algorithms both need an centralized entity to collect global network information, carry out the computation, and coordinate the implementation of the solution, making them less adaptive to system dynamics due to frequent reconstruction of trees.

Distributed solutions to the optimal broadcasting problem were then provided by Ho and Viswanathan [6] and by Zhang *et al.* [7], under the edge-capacitated network model and the node-capacitated network model, respectively. These solutions combine random network coding [6], [8] and back-pressure based capacity scheduling [9]. Although the algorithms are distributed, each participating user needs to *maintain one queue for every other user in the entire network*, which stores contents intended for that user. As the network size grows, the storage and communication overhead introduced by maintaining and updating all the queues at each user soon becomes prohibitive. Furthermore, upon user joins and departures, the whole network needs to be informed to add or remove queues, which is rather inefficient.

Massoulie *et al.* [10] proposed a simple distributed algorithm for both edge-capacitated and node-capacitated networks. Every user only needs to maintain one queue for each of its neighbors, which stores contents innovative to the corresponding neighbor. It is proved to support any feasible broadcast rate for arbitrary edge-capacitated networks and full-mesh node-capacity networks. But the algorithm fails when the capacity bottleneck is on the underlay links such as the example in Fig. 1. Also it works only if a feasible target broadcast rate is given. It *cannot approach the maximum broadcast rate* adaptively.

Despite all the existing results, the problem remains very challenging if we wish to achieve the maximum broadcast rate in a distributed fashion, while *maintaining only per-neighbor information queue at each node*. There is no existing problem formulation to help design such an algorithm. The problem becomes even harder when one considers adapting the algorithm to a general network model. Under our network model, the algorithm needs to learn underly capacity bottlenecks and adjust the overlay link rate accordingly.

In this work, we aim at designing such a broadcasting algorithm and make the following contributions:

1) We consider the model that the overlay network is acyclic but underlay capacity bottlenecks can be anywhere, which subsumes most known ones. Under this model, we formulate the optimal broadcasting problem in an innovative way. We show that our formulation is simple yet effective to characterize any feasible broad-

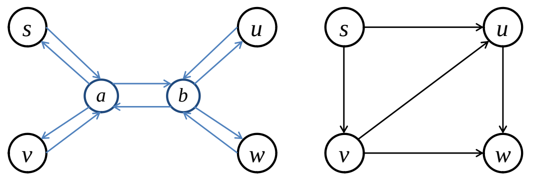


Fig. 1. Network Model: The graph on the right represents an overlay network built upon the network on the left. The broadcasting system works in the overlay network. Nodes a and b are routers; nodes s,u,v and w denote broadcast receivers. An edge is an physical link on the left, and is a multi-hop TCP/UDP connect on the right. For example, overlay link (v,u) traverses the physical path $(v,a) \to (a,b) \to (b,u)$. The capacity constraint can be anywhere — for example, on the overlay links (*edge-capacitated*) or on the broadcasting nodes (*node-capacitated*).

cast rate. Most importantly, this formulation leads us to efficient algorithm design.

2) Based on our formulation, we are the first to design a distributed algorithm to solve the optimal broadcasting problem, which requires only per-neighbor information maintained at each node. The proposed algorithm scales very well and is robust to high levels of network dynamics.

II. Problem Setting and Notations

A network is modeled as a directed graph $\mathcal{G} = (\mathcal{N},\mathcal{L})$, where $\mathcal{N}$ is the set of physical nodes including all the broadcasting participants and other intermediate nodes such as routers, and $\mathcal{L}$ is the set of all physical links.

Consider a single source broadcasting system deployed in an overlay network $G = (V,E)$ built upon $\mathcal{G}$. We assume that G is acyclic. The node set $V \in \mathcal{N}$ represents the set of all the broadcasting participants. The edge set E represents the set of *overlay* links connecting the nodes V. An *overlay* link is a *TCP* or *UDP* connection, which may traverse multiple physical links of $\mathcal{L}$. A simple example is shown in Fig. 1. Let $s \in V$ denote the source and $R = V - \{s\}$ denote the set of receivers. Let

$$in(v) = \{w \in V | (w,v) \in E\}$$

be the set of incoming neighbors of node v, from which node v may receive contents. Let

$$out(v) = \{u \in V | (v,u) \in E\}$$

be the set of outgoing neighbors of node v, to which node v can transmit contents. The source s generates contents continuously at rate z. Each receiver attempts to collect all the contents from its incoming neighbors.

Let r_{vu} be the content transmission rate over link $(v,u) \in E$. The network capacity constraints are expressed as follows:

$$A \cdot \boldsymbol{r} \leq \boldsymbol{C}, \tag{1}$$

where the column vector $\boldsymbol{r} = \{r_{vu}, (v,u) \in E\}$ denotes the link rate set, the column vector $\boldsymbol{C}$ denotes all the capacity bottlenecks in the network and the matrix A reflects how links share the network capacities. *Note that our network capacity model is rather general and the inequality (1) subsumes scenarios where the capacity bottleneck can be anywhere in the network.* For instance, for a P2P broadcasting system where the capacity constraint is generally assumed to be on the nodes, i.e., any node v's aggregate outgoing rate $\sum_{u\in out(v)} r_{vu}$ is upper bounded by the node v's upload capacity C_v. Then we can write $\boldsymbol{C} = \{C_v, v \in V\}$ and $A = \{a_{v,e}, v \in V, e \in E\}$ where

$$a_{v,e} = \begin{cases} 1 & \text{if the tail of } e \text{ is } v, \\ 0 & \text{otherwise.} \end{cases}$$

For example, in Fig. 1, for node v, we have $r_{vu} + r_{vw} \leq C_v$. While if the capacity constraint is on the physical links, for example, in wireless communication systems, we can write $\boldsymbol{C} = \{C_l, l \in \mathcal{L}\}$ and $A = \{a_{l,e}, l \in \mathcal{L}, e \in E\}$ where C_l is the link capacity and

$$a_{l,e} = \begin{cases} 1 & \text{if overlay link } e \text{ passes physical link } l, \\ 0 & \text{otherwise.} \end{cases}$$

For example, in Fig. 1, for physical link (a,b), we have $r_{su} + r_{vu} + r_{vw} \leq C_{ab}$. For the convenience of illustration, we use e and (v,u) exchangeably to denote one edge in E afterwards.

III. Problem Statement

Before presenting our problem, we first characterize the maximum broadcast rate B, i.e., the highest rate at which every node in the system can receive the contents simultaneously.

Given an $\boldsymbol{r} = \{r_{vu}, (v,u) \in E\}$ satisfying (1), for any $v \in R$, we let $\rho_{s,v}(\boldsymbol{r})$ be the minimum $s-v$ cut capacity. $\rho_{s,v}(\boldsymbol{r})$ can be expressed as follows

$$\begin{aligned} \rho_{s,v}(\boldsymbol{r}) = \quad & \min_{U,\bar{U}} \sum_{(u,w)\in E, u\in U, w\in \bar{U}} r_{uw} \\ \text{s.t.} \quad & s \in U,\ v \in \bar{U}, \\ & U \cup \bar{U} = V, \\ & U \cap \bar{U} = \phi, \end{aligned} \tag{2}$$

where the three constraints define that $(U,\bar{U})$ is a $s-v$ cut, and the objective is the cut capacity. By the well-known max-flow min-cut theorem, B is equal to the value of the following problem

$$\begin{aligned} & \min_{v\in R, \boldsymbol{r}} \rho_{s,v}(\boldsymbol{r}) \\ & \text{s.t. } A \cdot \boldsymbol{r} \leq \boldsymbol{C}. \end{aligned} \tag{3}$$

In other words, the maximum broadcast rate is equal to the minimum of the minimum source-receiver cut capacities across all feasible link rate allocations.

We wish to design a distributed algorithm to achieve the maximum broadcast rate B, and at the same time satisfy the requirement ***Neighbor* − *only***: *only neighbor-regarded information is maintained at each node*. For example, every node may keep each neighbor's current state (e.g., hitherto received contents) and update it periodically in the algorithm. Such information is generally used to guide node's behavior. The reason that only neighbor-regarded information is maintained is to guarantee good scalability and robustness. With only neighbor-regarded information, the storage and communication overhead is lightweight even in large-scale systems. Upon node joins/departures, we only need to notify the corresponding neighbors. Consequently, even if the system is highly dynamic, the algorithm can still adapt in an agile and efficient way. We will further explain the requirement ***Neighbor* − *only*** and its benefits in Section V-B when we describe our broadcasting algorithm.

IV. Problem Formulation and Justification

A. Problem Formulation

The optimal broadcasting problem can be formulated in different ways. Besides (3), the information-flow based formulation is most common [6], [7]. However, none of the algorithms based on existing formulations can satisfy the strict requirement of ***Neighbor − only***. Our new formulation below enables us to design a broadcasting algorithm that is throughput optimal and distributed, and most importantly, satisfies the ***Neighbor − only*** condition.

We formulate the broadcast rate maximization problem as follows:

$$\max_{z\geq 0, \boldsymbol{r}\geq 0} U(z) \tag{4}$$

$$\text{s.t.} \sum_{u\in in(v)} r_{uv} \geq \sum_{p\in in(w)} r_{pw} + z\mathbf{1}_{w=s}, \forall v\in R, w\in in(v), \tag{5}$$

$$A\cdot \boldsymbol{r} \leq \boldsymbol{C}, \tag{6}$$

where z is the rate at which the source s generates contents, $U(\cdot)$ is a differentiable strictly concave increasing utility function, and r_{uv} is the allocated rate over overlay link (u, v) at which node u can transmit contents to node v. Constraint (5) states that a node's total receiving rate should be no less than the total receiving rate of each of its incoming neighbors. The network capacity constraint is captured by constraint (6).

Constraint (5) differentiates our problem formulation fundamentally from existing ones in the literature. Prior to this work, the information-flow based formulation is mostly used to help design broadcasting algorithms. In the information-flow based formulation, instead of (5), there is a flow-balancing constraint for *every node pair*. In our formulation, however, (5) involves *only node-neighbor pairs*. This unique feature guarantees that, in our broadcasting algorithm based on this formulation, every node interacts just with its neighbors and only neighbor-regarded information is needed.

Now we explain roughly why the constraint (5) makes sense. It is known that, for any acyclic graph, we can always divide the nodes into different ordered layers such that the source is exclusively in the lowest layer and every node stays in lower layer than any of its outgoing neighbors. In other words, nodes can only receive contents from those in lower layers. Let's consider nodes layer by layer from bottom to top. First, nodes in the second lowest layer can receive contents only from the source. The constraint (5) makes sure that these nodes can obtain whatever the source has, since the receiving rate is no less than z. Next, nodes in the third lowest layer can receive the contents from either the source or nodes in the second lowest layer. Due to the similar reason, the constraint (5) guarantees these nodes can also get all the source has. Then we can continue the same argument until the highest layer. So essentially the constraint (5) is to make every receiver receive all the contents generated by the source.

Next, we formally show that the simple constraint (5) along with (6) give us the feasible region of the broadcast rate for any acyclic graph.

B. Formulation Justification

In this subsection, we first establish a known property about the acyclic graph G: nodes can be indexed so that every node's index is smaller than that of any outgoing neighbor. Then, based on this property, we show that constraints (5), (6) are sufficient and necessary conditions for any feasible broadcast rate $z \leq B$.

The topological ordering of acyclic graphs is a known result; it is presented in the following proposition for the sake of completeness.

Proposition 1: All nodes in G can be sequentially indexed such that, for any node $v \in R$, the index of any $u \in in(v)$ is smaller than v's index.

We say a broadcast rate z is feasible if and only if $z \leq B$. Based on Proposition 1, we show in the following theorem that constraints (5), (6) are sufficient and necessary conditions for any feasible broadcast rate under the acyclic directed graph. To show necessity, we prove that, for any feasible z, we can find a $\boldsymbol{r}$ that supports z and at the same time satisfies constraints (5), (6). On the other hand, for any $s - v$ cut in G, we can always find a node such that the cut capacity is no less than the node's total receiving rate. Constraint (5) guarantees that every node's receiving rate is larger than or equal to the broadcast rate z. So the broadcast rate z is not larger than any cut capacity, and thus feasible by the definition of B in (3).

Theorem 1: If G is acyclic, any z such that there exists a $\boldsymbol{r} = \{r_e, e \in E\}$ satisfying the constraints (5), (6) is a feasible broadcast rate. On the other hand, for any feasible broadcast rate z, there exists one $\boldsymbol{r} = \{r_e, e \in E\}$ satisfying the constraints (5), (6).

The proof can be found in [11].

V. Algorithm Design

Based on our problem formulation, we now apply the classic Lagrangian decomposition approach to design a broadcasting algorithm, which is throughput optimal, distributed and maintains only neighbor-regarded information at each node.

A. Lagrangian Decomposition

By relaxing constraint (5), we obtain the following partial Lagrangian:

$$\begin{aligned} & L(z, \boldsymbol{\theta}, \boldsymbol{r}) \\ =& U(z) + \sum_{v\in R}\sum_{w\in in(v)} \theta_{v,w}\left(\sum_{u\in in(v)} r_{uv} - \sum_{p\in in(w)} r_{pw} - z\mathbf{1}_{w=s}\right), \end{aligned}$$

where $\theta_{v,w}$ is the Lagrange multiplier.

Since the Slater constraint qualification conditions hold for the problem (4) [12], strong duality holds. Thus problem (4) can be solved by finding the saddle points of $L(z, \boldsymbol{\theta}, \boldsymbol{r})$, through solving the following problem in $z, \boldsymbol{\theta}, \boldsymbol{r}$:

$$\begin{aligned} \min_{\boldsymbol{\theta}\geq 0} \Bigg(& \max_{z\geq 0}\left[U(z) - \sum_{v\in R: s\in in(v)} \theta_{v,s} z\right] + \\ & \max_{\boldsymbol{r}\geq 0} \sum_{v\in V}\sum_{u\in out(v)} r_{vu}\left[\sum_{w\in in(u)} \theta_{u,w} - \sum_{w\in out(u)} \theta_{w,u}\right]\Bigg) \\ & \text{s.t.} A\cdot \boldsymbol{r} \leq \boldsymbol{C}. \end{aligned}$$

Given $\boldsymbol{\theta}$, we first solve the following capacity scheduling subproblem in $\boldsymbol{r}$:

$$\textbf{SSP}: \max_{\boldsymbol{r}\geq 0} \sum_{v\in V}\sum_{u\in out(v)} r_{vu}\left[\sum_{w\in in(u)}\theta_{u,w} - \sum_{w\in out(u)}\theta_{w,u}\right]$$
$$\text{s.t.} A\cdot \boldsymbol{r} \leq \boldsymbol{C}.$$

We can exploit the specific structure of the above linear program to solve it in a distributed fashion. In particular, for any two neighboring nodes u and v, we define the back-pressure from u to v as

$$P_{vu} = \left[\sum_{w\in in(u)}\theta_{u,w} - \sum_{w\in out(u)}\theta_{w,u}\right], \ \forall (v,u)\in E. \qquad (7)$$

Under different scenarios of network capacity constraints, we solve this subproblem correspondingly, and obtain the optimal solution $\boldsymbol{r}^*$.

If capacity constraints are applied on the nodes:

$$\sum_{u\in out(v)} r_{vu} \leq C_v, \ \forall v\in V,$$

for any $v\in V$, let $u^*(v) = \arg\max_{u\in out(v)} P_{vu}$, then the solution to **SSP** is as follows: for any $v\in V$,

$$r^*_{vu} = \begin{cases} C_v, & \text{if } u\in out(v),\ u = u^*(v) \text{ and } P_{vu} > 0,\\ 0, & \text{otherwise.}\end{cases} \qquad (8)$$

If capacity constraints are applied on the physical links, recall that they can be written as:

$$\sum_{e\in E} a_{l,e} r_e \leq C_l, \ \forall l\in \mathcal{L},$$

where $\mathcal{L}$ is the set of all physical links and

$$a_{l,e} = \begin{cases} 1 & \text{if overlay link } e \text{ uses physical link } l,\\ 0 & \text{otherwise.}\end{cases}$$

We can solve **SSP** by the following primal-dual algorithm which will converge to $\boldsymbol{r}^*$

$$\begin{cases} \dot{r}_e = \beta_e\left[P_e - \sum_{l\in\mathcal{L}} a_{l,e}\lambda_l\right]^+_{r_e}, \ \forall e\in E,\\ \dot{\lambda}_l = \sigma_l\left[\sum_{e\in E} a_{l,e} r_e - C_l\right]^+_{\lambda_l}, \ \forall l\in\mathcal{L},\end{cases} \qquad (9)$$

where β_e and σ_l are positive constants, and function

$$[b]^+_a = \begin{cases} \max(0,b) & a\leq 0\\ b & a > 0.\end{cases}$$

Given the **SSP**'s solution r^*_{vu} as (8) or (9), we use the following distributed primal-dual algorithm to solve the sub-problems in $z, \boldsymbol{\theta}$ simultaneously:

$$\begin{cases} \dot{z} = \alpha\left[U^{'}(z) - \sum_{v\in V: s\in in(v)}\theta_{v,s}\right]^+_z,\\ \dot{\theta}_{v,u} = \gamma_{v,u}\Big[\sum_{p\in in(u)} r^*_{pu} + z\mathbf{1}_{u=s} \\ \qquad - \sum_{w\in in(v)} r^*_{wv}\Big]^+_{\theta_{v,u}}, \ \forall v\in R, u\in in(v),\end{cases} \qquad (10)$$

where α and $\gamma_{v,u}$ are positive constants.

B. Broadcasting Algorithm

Our broadcasting algorithm works as follows: Every node v maintains one "queue" $\theta_{v,u}$ for each incoming neighbor $u\in in(v)$. In each time slot,

- **Primal-dual Update**: According to (10), the source s updates the broadcast rate z, and each node v collects information from $u\in in(v)$ and updates the queue $\theta_{v,u}$.
- **Capacity Scheduling**: Node v decides the link transmission rate r_{vu} for all $u\in out(v)$ according to (8) or (9).
- **Content Scheduling**: Given $\boldsymbol{r}$, every node v coordinates with its incoming neighbor set $\{u | u\in in(v), r_{uv} > 0\}$, and decides what to receive from each of them in order to obtain as many innovative contents as possible. Then every node u sends out specific contents to each outgoing neighbor $v\in out(u)$ at rate r_{uv}.

Remark: We can also adopt random network coding to help content scheduling [8], [13]. In each time slot, every node u randomly encodes all the received contents and sends out the coded contents to each $v\in out(u)$ at rate r_{uv}. This way, there is no need for coordination between each node-neighbor pair. However, network coding will introduce communication overhead for carrying coding coefficients and computation complexity for encoding and decoding.

We have the following observations for our broadcasting algorithm:

- The Lagrangian variable $\theta_{v,u}$ measures the buffer size difference between node v and its incoming neighbor u, which can be calculated at node v by collecting information from each of its neighbors.
- With the above understanding on $\boldsymbol{\theta}$, the terms in P_{vu} can be understood as follows. The term $\sum_{w\in in(u)}\theta_{u,w}$ measures the aggregate deficit in received content amount between node u and all its incoming neighbors. The larger this term is, the more desperate peer u wants to receive contents. Similarly, the term $\sum_{w\in out(u)}\theta_{w,u}$ measures the aggregate surplus in received content amount between node u and all its outgoing neighbors.
- The algorithm requires nodes to exchange information only with its one-hop neighbors, and can be implemented in a distributed manner. At each node, the maintained information is $\theta_{v,u}$ regarding only incoming neighbors. Hence our algorithm satisfies the $\boldsymbol{Neighbor-only}$ condition. The number of "queues" every node needs to maintain and update is just equal to half of the size of its neighbors. As a result, the storage and communication cost of maintaining and updating jobs at each node is quite limited even for large-scale systems. When there is a node departing, that node just needs to notify its outgoing neighbors, who eliminate the "queue" regarding the leaving node. When there is a new node joining, that node first establishes one "queue" for each incoming neighbor, and then notifies its outgoing neighbors. Each outgoing neighbor adds one "queue" regarding the joining node. The adjustment overhead incurred by network dynamic is only proportional to the neighbor size and thus very lightweight. Our algorithm, therefore, scales very well and is very robust to network dynamics.

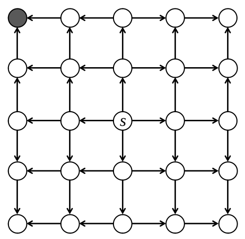


Fig. 2. Topology of the simulated overlay network: The center node is the source. The graph has a "grid" topology. When the number of nodes increases, the graph expands symmetrically as a 2D grid. The dark node is the broadcast bottleneck.

Next, we show that our broadcasting algorithm is throughput optimal. This means that our algorithm can guarantee high performance, for example, high-quality video support, in practical systems. High performance can improve user experience and attract more users in return.

To show this, we proceed in two steps. First, we show that the joint primal-dual algorithm and capacity scheduling can converge to the optimal solution z^*, $\boldsymbol{r}^*$of (4). Then, under rate allocation $\boldsymbol{r}^*$, our content scheduling strategy can guarantee that every node can receive innovative contents at rate z^*. By Theorem 1, we get $z^* = B$ which is the maximum broadcast rate. Overall, we have the following theorem.

Theorem 2: Our broadcasting algorithm is throughput optimal.

The proof can be found in [11].

VI. Simulation

We implement our broadcasting algorithm using Python and conduct simulation studies to evaluate the performance of our solution.

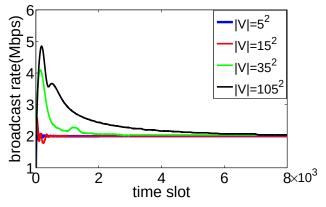
A. Settings

In our simulations, time is chopped into slots of equal length. The topology of the overlay network is shown in Fig. 2. We adopt two different settings. In Setting I, the capacity constraint is on the edge. The edge capacity is 4 Mbps except the incoming edges of the top-left node (filled in black in Fig. 2), at 1 Mbps each. In Setting II, the capacity constraint is on the nodes. The node capacity is 8 Mbps except the source and the incoming neighbors of the top-left node. The source's capacity is 16 Mbps. The capacity of each incoming neighbor of the top-left node is 1 Mbps. In both settings, given our capacity constraints, the broadcast bottleneck is at the network edge and thus far from the source. It's easy to check that the maximum broadcast rate under both settings is 2 Mbps. We choose the number of nodes as $5^2, 15^2, 35^2, 105^2$ respectively.

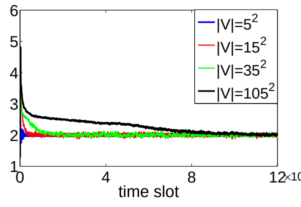
Under Setting I and II, we next evaluate our broadcasting algorithm.

B. Evaluation of the Proposed Broadcasting Algorithm

We evaluate our broadcasting algorithm proposed in Section V-B and answer the following questions: 1) does it converge to the maximum broadcast rate as expected by the theoretical analysis? 2) how fast does it converge? 3) what's the impact of network size on the convergence time (the length of the interval from the start to where the broadcast rate begins to stay with the optimum)?



(a) Broadcasting under Setting I



(b) Broadcasting under Setting II

Fig. 3. Simulation results

We show the results under Setting I in Fig. 3(a) and the results under Setting II in Fig. 3(b). We have the following observations. First, our broadcasting algorithm can converge to the maximum broadcast rate 2 Mbps and is thus optimal under different network capacity models. Second, when the network grows, the convergence time increases. This is because the bottleneck (at the network edge) is further from the source as the number of nodes gets larger. It takes longer from the source to learn about the bottleneck and to adjust the broadcast rate correspondingly. Third, under Setting I, the convergence times are about $300, 600, 2000, 6000$ time slots respectively as the number of nodes increases; under Setting II, the convergence times are $1000, 5000, 19000, 10^5$ time slots respectively. We can see that the convergence time of our algorithm grows linearly with the network size under both settings. That means our algorithm is suitable to be implemented in large-scale systems. It might be an interesting future direction to theoretically analyze the convergence behavior of our algorithm. Forth, the convergence under Setting II is longer than that under Setting I. The reason is that, different from Setting I, each node needs to allocate its upload capacity among its neighbors every time slot under Setting II.

References

[1] PPLive. http://www.pplive.com/.

[2] UUSee. http://www.uusee.com/.

[3] J. Edmonds, "Edge-disjoint branchings," *Combinatorial Algorithms, R. Rustin, ed.*, 1973.

[4] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. A. Chou, "Peer-to-peer streaming capacity," *IEEE Trans. Information Theory*, 2011.

[5] N. Garg and J. Konemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *IEEE Symp. Foundations of Computer Science*, 1998.

[6] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *IEEE Trans. Information Theory*, 2005.

[7] S. Zhang, Z. Shao, and M. Chen, "Optimal distributed p2p streaming under node degree bounds," in *Proc. IEEE ICNP*, 2010.

[8] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conference*, 2003.

[9] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, 1992.

[10] L. Massoulie, A. Twigg, G. Gkantsidis, and P. Rodriguez, "Randomized Decentralized Broadcasting Algorithms," in *Proc. IEEE INFOCOM*, 2007.

[11] S. Zhang, M. Chen, Z. Li, and L. Huang, "Optimal distributed broadcasting with per-neighbor queue in acyclic overlay networks with arbitrary underlay capacity constraints," *Technical Report*, available at http://arxiv.org/abs/1301.5107.

[12] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[13] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Information Theory*, 2004.