

Optimal Distributed P2P Streaming under Node Degree Bounds

Shaoquan Zhang, Ziyu Shao, and Minghua Chen

Department of Information Engineering, The Chinese University of Hong Kong

Email: {zsqu008, zyshao6, minghua}@ie.cuhk.edu.hk

Abstract—We study the problem of maximizing the broadcast rate in peer-to-peer (P2P) systems under *node degree bounds*, i.e., the number of neighbors a node can simultaneously connect to is upper-bounded. The problem is critical for supporting high-quality video streaming in P2P systems, and is challenging due to its combinatorial nature. In this paper, we address this problem by providing the first distributed solution that achieves near-optimal broadcast rate under arbitrary node degree bounds, and over arbitrary overlay graph. It runs on individual nodes and utilizes only the measurement from their one-hop neighbors, making the solution easy to implement and adaptable to peer churn and network dynamics. Our solution consists of two distributed algorithms proposed in this paper that can be of independent interests: a network-coding based broadcasting algorithm that optimizes the broadcast rate given a topology, and a Markov-chain guided topology hopping algorithm that optimizes the topology. Our distributed broadcasting algorithm achieves the optimal broadcast rate over arbitrary P2P topology, while previously proposed distributed algorithms obtain optimality only for P2P complete graphs. We prove the optimality of our solution and its convergence to a neighborhood around the optimal equilibrium under noisy measurements or without time-scale separation assumptions. We demonstrate the effectiveness of our solution in simulations using uplink bandwidth statistics of Internet hosts.

I. INTRODUCTION

Peer-to-peer (P2P) systems have provided a scalable and cost effective way for streaming video in the past decade. Recent studies [11]–[14], however, indicate that the practical performance of P2P streaming systems can be far from their theoretical optimal.

There have been work studying the performance limit of P2P systems to understand and unleash their potential. One focus is on the *streaming capacity* problem [15], i.e., maximizing the streaming rate subject to the peering and overlay topology constraints. The problem is critical for supporting high-quality video, which is determined by the streaming rate, in P2P systems. In this paper, we focus on the broadcast scenario where all peers in the system are receivers.

The case of unconstrained peering on top of a complete graph is well studied, where the maximum broadcast rate is derived in several papers [1]–[3], [16], [17]. The case of unconstrained peering over general graph can also be addressed by using a centralized solution [5].

The streaming capacity problem becomes NP-Complete over general graph with *node degree bounds* [10]. Node degree is defined as the number of neighbors a node maintains connections to. Due to connection overhead costs, it is necessary

to limit the number of simultaneous connections a peer can maintain. This naturally bounds the node degrees in P2P systems.

There has been work studying this challenging problem. SplitStream/CoopNet [6], [7], ZIGZAG [8], PRIME [9] and most practical systems (such as PPLive [18] and UUSee [19]) bound node degree but do not provide rate optimality guarantee. Recently, the authors in [10] proposed a centralized Cluster-Tree algorithm that achieves near-optimal broadcast rate with high probability over complete graph, under the assumption that the node degree bound is at least logarithmic in the size of the network. A summary and comparison of previous work and this work are in Table I.

Despite of these exciting results, the following two important questions remain open:

- What is the maximum broadcast rate under arbitrary node degree bounds, and over general P2P overlay graph?
- How to achieve the maximum broadcast rate in a *distributed* manner?

Systems running distributed algorithms, compared with those running centralized algorithms, are more adaptable to peer churn and network dynamics.

In this paper, we answer the above two questions and make the following contributions:

- We provide the first distributed solution that achieves a broadcast rate arbitrarily close to the optimal under arbitrary node degree bounds, and over arbitrary overlay graph. Our solution runs on individual nodes and utilizes only the information from their one-hop neighbors.

Our solution consists of the following two algorithms that can be of independent interests.

- We propose a distributed broadcasting algorithm that achieves the optimal broadcast rate over arbitrary overlay graph. Previous distributed P2P broadcasting algorithms are optimal only for complete overlay graph [1]–[3]. Our algorithm is based on network coding and utilizes back-pressure arguments.
- We also propose a distributed algorithm that optimizes the topology. In this algorithm, each node hops among their possible set of neighbors towards the best peering configuration. Our algorithm is inspired by a set of log-sum-exp approximation and Markov chain based arguments expounded in [20].
- We prove the optimality of the overall solution. We also

TABLE I
SUMMARY AND COMPARISON OF PREVIOUS WORK AND THIS WORK FOR MAXIMIZING P2P BROADCAST RATE.

References	General Overlay Graph?	Arbitrary Node Degree Bound?	Exact or $1 - \epsilon$ Optimality?	Distributed Solution?
Mutualcast [1] and the algorithms in [2], [3]	×	×	✓	✓
Iterative in [4], [5]	✓	×	✓	×
CoopNet/SplitStream [6], [7]	×	✓	×	×
ZIGZAG [8], PRIME [9]	✓	✓	×	✓
Cluster-tree [10]	×	✓	conditionally optimal *	×
This paper	✓	✓	✓	✓

* The Cluster-Tree algorithm is $(1 - \epsilon)$ -optimal with high probability if the node degree bound is $O(\log N)$.

prove its convergence to a neighborhood around the optimal equilibrium in the presence of noisy measurements or without time-scale separation assumptions. We demonstrate the effectiveness of our solution in simulations using uplink bandwidth statistics of Internet hosts.

All proofs and pseudo codes are in [21].

II. PROBLEM FORMULATION

A. Settings and Notations

We model the P2P overlay network as a general directed graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of links. Each link in the graph corresponds to a TCP/UDP connection between two nodes. Each node $v \in V$ is associated with an upload capacity $C_v \geq 0$. We assume there is no constraint on the downloading rate for each node $v \in V$. This assumption can be partly justified by the empirical observation that as residential broadband connections with asymmetric upload and download rates become increasingly dominant, bottlenecks typically are at the uplinks of the access networks rather than in the middle of the Internet.

As such, P2P networks have capacity limits on the nodes instead of links. This is different from traditional underlay networks where the capacity limits are on the links.

We focus on the single-source streaming scenario, i.e., a source s broadcasts a continuous stream of contents to the entire network; we denote its receiver set as $R \triangleq V - \{s\}$.

We consider the peering constraints that each node has a degree bound B_v , i.e., it can only *simultaneously* connect to a B_v number of neighbors due to connection overhead cost. We allow different nodes to have different degree bounds. Fig. 1 shows four sample peering configurations of a 5-node network with node degree bound 3 for each node.

Let \mathcal{F} denote the set of all feasible peering configurations over graph G under node degree bounds. Given a configuration $f \in \mathcal{F}$, we obtain a connected sub-graph of G that satisfies the node degree bound constraints. We denote this sub-graph as $G_f = (V, E_f)$, where E_f represents the set of links in this sub-graph. We denote $N_{v,f}$ as the set of node v 's neighbors in this sub-graph. We have $|N_{v,f}| \leq B_v$ where $|\cdot|$ represents the size of a set.

B. Problem Formulation and Our Approach

For a configuration $f \in \mathcal{F}$, let x_f be the maximum achievable broadcast rate under f . The problem of maximizing

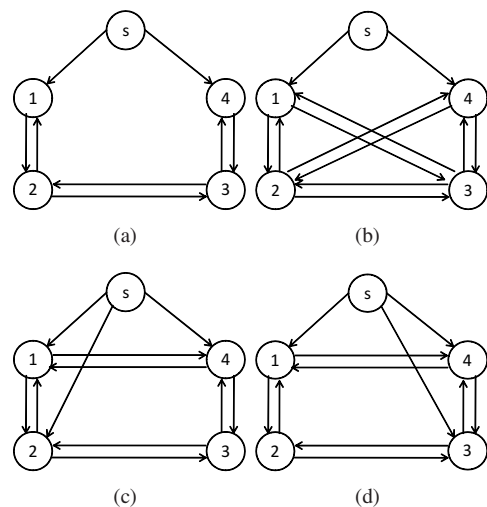


Fig. 1. Peering configuration examples for a 5-node network with node degree bound 3 for each node.

broadcast rate under node degree bounds can be formulated as follows:

$$\text{MRC} : \max_{f \in \mathcal{F}} x_f. \quad (1)$$

This problem is known to be NP-complete [10], and there is no effective approximate solution to the problem even in a centralized manner.

In this paper, we address this problem by providing a distributed solution. In particular, we first develop a distributed broadcasting algorithm that can achieve x_f under arbitrary $f \in \mathcal{F}$. We then design a distributed algorithm that optimizes towards the best peering configurations. They operate in tandem to achieve a close-to-optimal broadcast rate under arbitrary node degree bounds, and over arbitrary overlay graph. We elaborate on these two algorithms in the following two sections.

III. THE PROPOSED DISTRIBUTED BROADCASTING ALGORITHM

By exploiting network coding [22], we design a back-pressure based distributed broadcasting algorithm. This algorithm can achieve the maximum broadcast rate over arbitrary P2P topology.

A. Routing vs. Network Coding

In P2P systems, there are two approaches for broadcasting contents: one is based on routing, in which nodes only store and forward packets; and the other is based on network coding [22], in which a node is also allowed to mix information and output data as functions of the data it received. Some commercial P2P systems are built upon routing-based approach (e.g., PPLive [18]), and some are based on network coding (e.g., UUSee [19], [23])¹. It is known that both routing and network coding approaches can achieve optimal broadcast rate over arbitrary P2P graph [2], [17]. The routing-based approaches, however, are not robust as they usually incur substantial overhead in constructing and maintaining the spanning trees in the presence of peer churn and system dynamics. In this section, we design a distributed broadcasting algorithm based on network coding that is robust to dynamics.

B. Network Coding Based Formulation

According to the Max-Flow-Min-Cut theorem, a data transmission of rate z between source s and a receiver d is *feasible* if and only if there exists a flow, denoted as f^d , satisfying the following flow conservation constraints:

$$\sum_{u \in \text{in}(v)} f_{uv}^d \leq \sum_{u \in \text{out}(v)} f_{vu}^d, \quad \forall v \in R - \{d\}, \quad (2)$$

$$z \leq \sum_{u \in \text{out}(s)} f_{su}^d, \quad (3)$$

$$0 \leq f^d, \quad (4)$$

where $\text{in}(v) \triangleq \{u | (u, v) \in E_f\}$ is the set of nodes sending content to v under configuration f , and $\text{out}(v) \triangleq \{u | (v, u) \in E_f, u \neq s\}$ is the set of nodes receiving content from v .

A powerful theorem established in [22] states that a multicast or broadcast rate z from s to a set of receivers is achievable if and only if z is feasible for s and any receiver d . This is a strong result as it says that if the network can support a unicast rate of z between s and any receiver assuming other receivers' traffic is absent, then it can support a multicast rate of z to all the receivers simultaneously. Such rate z can be achieved by every node in the network performing network coding [22]. Further, authors in [25], [26] show that it is sufficient to perform random linear network coding.

In random linear network coding, by independently and randomly choosing a set of coding coefficients from a finite field, each node sends out the coded packet as a linear combination of the node's received packets. The combination information is specified by a *coefficient vector* in the packet header, which is updated by applying the same linear transformations as to the data. When one node receives a full set of linearly independent coded packets, it can decode and recover the original packets. In this paper, we focus on the distributed algorithm design.

¹We refer interested readers to [23], [24] for more details on performance of routing-based and network-coding-based practical P2P systems. We focus on optimal distributed P2P broadcasting algorithm design based on network coding in this paper.

The discussions of decoding probability and implementation details can be found in [25], [26].

Under the setting of network coding, we can consider f^d as a "virtual" information flow between s and d . Multiple information flows "piggyback" together to transmit over the physical links. The actual physical rate over a physical link is only the maximum rate of individual information flows passing over it. Let g_{uv} be the physical flow rate over a link $(u, v) \in E_f$, then we have $f_{uv}^d \leq g_{uv}$ for all $d \in R$.

With the above understanding, we formulate the problem of maximizing broadcast rate under configuration f as follows:

$$\mathbf{MP} : \max_{z, f, g \geq 0} U(z) \quad (5)$$

$$\text{s.t.} \quad \sum_{u \in \text{in}(v)} f_{uv}^d + z \mathbb{1}_{\{v=s\}} \leq \sum_{u \in \text{out}(v)} f_{vu}^d, \quad \forall v \in V - \{d\}, d \in R \quad (6)$$

$$f_{vu}^d \leq g_{vu}, \quad \forall v \in V, \forall u \in \text{out}(v), d \in R, \quad (7)$$

$$\sum_{u \in \text{out}(v)} g_{vu} \leq C_v, \quad \forall v \in V, \quad (8)$$

where $U(z)$ is a twice-differentiable strictly concave utility function², $\mathbb{1}_{\{t\}}$ denotes the indicator function. The constraints in (6) describe the flow conservation requirements. The constraints in (7) come from the piggybacking property of information flows. The node upload capacity constraints are in (8). The problem **MP** is a convex problem. All feasible broadcast rates must satisfy the constraints in (6)-(8) and are achievable by using random linear network coding.

C. Algorithm Design via Lagrange Decomposition

To proceed, we first relax the first set of constraints in (6) in problem **MP** to obtain a partial Lagrangian as follows:

$$\begin{aligned} L(z, f, g, \lambda) &= U(z) - \sum_{v \in V - \{d\}} \sum_{d \in R} \lambda_{v,d} \left(\sum_{u \in \text{in}(v)} f_{uv}^d + z \mathbb{1}_{\{v=s\}} - \sum_{u \in \text{out}(v)} f_{vu}^d \right) \\ &= U(z) - \sum_{v \in V} \sum_{d \in R} \lambda_{v,d} \left(\sum_{u \in \text{in}(v)} f_{uv}^d + z \mathbb{1}_{\{v=s\}} - \sum_{u \in \text{out}(v)} f_{vu}^d \right) \end{aligned} \quad (9)$$

where $\lambda_{v,d}, v \in V - \{d\}, d \in R$ are Lagrange multipliers, $\lambda_{d,d} = 0, \forall d \in R$, and $\sum_{u \in \text{in}(s)} f_{us}^d = 0$.

The strong duality holds for problem **MP** since the Slater conditions are satisfied [27]. Therefore, we can solve problem **MP** by finding the saddle points of $L(z, f, g, \lambda)$.

Noticing that

$$\sum_{v \in V} \sum_{d \in R} \lambda_{v,d} z \mathbb{1}_{\{v=s\}} = z \sum_{d \in R} \lambda_{s,d} \quad (10)$$

and

$$\sum_{v \in V} \sum_{d \in R} \lambda_{v,d} \left(\sum_{u \in \text{in}(v)} f_{uv}^d - \sum_{u \in \text{out}(v)} f_{vu}^d \right) = \sum_{d \in R} \sum_{v \in V} \sum_{u \in \text{out}(v)} f_{vu}^d (\lambda_{u,d} - \lambda_{v,d}), \quad (11)$$

²It might seem unnecessary to involve a strictly concave utility function in this formulation. The reason is that we later design a primal-dual algorithm to solve the problem, and using a strictly concave utility function can avoid its potential instability problem [17].

we can find the saddle points of $L(z, \mathbf{f}, \mathbf{g}, \boldsymbol{\lambda})$ by solving the following problem successively in $z, \mathbf{f}, \mathbf{g}, \boldsymbol{\lambda}$:

$$\min_{\boldsymbol{\lambda} \geq \mathbf{0}} \left(\max_{z \geq 0} (U(z) - z \sum_{d \in R} \lambda_{s,d}) + \max_{\mathbf{f}, \mathbf{g} \geq \mathbf{0}} \sum_{d \in R} \sum_{v \in V} \sum_{u \in \text{out}(v)} f_{vu}^d (\lambda_{v,d} - \lambda_{u,d}) \right) \quad (12)$$

s.t. (7) – (8).

Given $\boldsymbol{\lambda}$ and z , we consider the following scheduling sub-problem on \mathbf{f}, \mathbf{g} :

$$\text{SSP} : \max_{\mathbf{f}, \mathbf{g} \geq \mathbf{0}} \sum_{d \in R} \sum_{v \in V} \sum_{u \in \text{out}(v)} f_{vu}^d (\lambda_{v,d} - \lambda_{u,d}) \quad (13)$$

s.t. (7) – (8).

The above linear programming problem has a structure that allows us to solve it distributedly. The first observation is that if an optimal \mathbf{g}^* is given, then an optimal \mathbf{f}^* can be obtained as follows: $\forall u, v \in V, d \in R$,

$$(f_{vu}^d)^* = \begin{cases} 0, & \text{if } \lambda_{v,d} - \lambda_{u,d} \leq 0, \\ g_{vu}^*, & \text{otherwise.} \end{cases} \quad (14)$$

As such, it is sufficient to study the following problem in \mathbf{g} :

$$\max_{\mathbf{g} \geq \mathbf{0}} \sum_{v \in V} \sum_{u \in \text{out}(v)} g_{vu} w_{vu} \quad (15)$$

s.t. $\sum_{u \in \text{out}(v)} g_{vu} \leq C_v, \forall v \in V$

where

$$w_{vu} \triangleq \sum_{d \in R} [\lambda_{v,d} - \lambda_{u,d}]^+, \quad \forall (u, v) \in E_f. \quad (16)$$

denotes the aggregate back-pressure between two neighboring nodes u and v , and $[\cdot]^+ \triangleq \max(\cdot, 0)$.

For any $v \in V$, let

$$u^*(v) \triangleq \arg \max_{u \in \text{out}(v)} w_{vu} \quad (17)$$

be one of its neighbors with the maximum back-pressure (breaking ties arbitrarily). Then one optimal solution for problem **SSP** is as follows:

$$(g_{vu}^d)^* = \begin{cases} 0, & \text{if } w_{vu} \geq 0 \text{ and } u = u^*(v), \\ C_v, & \text{otherwise,} \end{cases} \quad (18)$$

and

$$(f_{vu}^d)^* = \begin{cases} 0, & \text{if } \lambda_{v,d} - \lambda_{u,d} \leq 0, \\ g_{vu}^*, & \text{otherwise.} \end{cases} \quad (19)$$

Given \mathbf{f}^* and \mathbf{g}^* , primal-dual algorithms can be designed to adapt z and $\boldsymbol{\lambda}$ to pursue the desired optimal solution.

We summarize the above analysis into a distributed algorithm including the following components:

Primal-dual Rate Control: we pursue the saddle point in z and $\boldsymbol{\lambda}$ simultaneously as follows:

$$\begin{cases} \dot{z} = \alpha [U'(z) - \sum_{d \in R} \lambda_{s,d}]_z^+, \\ \dot{\lambda}_{v,d} = k_{v,d} \left[\sum_{u \in \text{in}(v)} (f_{uv}^d)^* + z \mathbb{1}_{v=s} - \sum_{u \in \text{out}(v)} (f_{vu}^d)^* \right]_{\lambda_{v,d}}^+, \quad \forall v \in V - \{d\}, d \in R, \\ \dot{\lambda}_{d,d} = \lambda_{d,d} = 0, \quad \forall d \in R, \end{cases} \quad (20)$$

where α and $k_{v,d}$ are positive step sizes, and the function

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

Neighbor Scheduling, Content Scheduling, and Network Coding: Every node $v \in V$ maintains a queue storing packets that are intended for d . Whenever a transmission opportunity arises, node v chooses one neighbor $u^*(v)$ with the maximum back-pressure according to (17).

If $w_{vu^*(v)} > 0$, node v sends packets to $u^*(v)$ at rate C_v . Every output packet is constructed as follows. Node v chooses one packet from the head of each queue of d if $\lambda_{v,d} - \lambda_{u^*(v),d} > 0$, and output one random linear combination of these head-of-queue packets. If otherwise $w_{vu^*(v)} \leq 0$ or there is no head-of-line packets to code, node v does nothing.

We have the following observations.

- The Lagrangian variable $\lambda_{v,d}$ is proportional to the length of queue storing packets that are intended for receiver d . The back-pressure w_{vu} measures the aggregate difference in the queues of all $d \in R$ between v and u . The larger the back-pressure is, the more desperate node u wants to receive data from v .
- Our algorithm can be implemented in a distributed manner. It only requires nodes to exchange information with its one-hop neighbors, and thus is robust to peer churn and system dynamics.
- Although our algorithm is designed for P2P broadcast scenarios, it also works for P2P multicast scenarios where helper nodes exist. The helper nodes simply also perform the operations described in (18)-(20). Our algorithm can be considered as the extension of the algorithm in [25] from link-capacity-limited underlay networks to node-capacity-limited overlay networks.

The following theorem characterize the convergence of the proposed algorithm.

Theorem 1: The algorithm in (18)-(20) converges to the optimal solution of problem **MP** globally asymptotically.

The proof utilizes standard Lyapunov arguments and a Lyapunov function for primal-dual algorithm, similar to those used in [17], [28].

Remark: We derive our algorithm and prove its convergence based on a fluid model formulation. It is also possible to obtain a similar back-pressure based distributed algorithm with packet-level dynamics taken into account and prove its stability, following a set of Lyapunov drift arguments elaborated in [29].

IV. THE PROPOSED DISTRIBUTED TOPOLOGY HOPPING ALGORITHM

Following the Markov approximation framework in [20], we design a topology hopping algorithm to optimize the peering configuration in a distributed manner. There are two steps in designing our algorithm under the Markov approximation framework [20]: log-sum-exp approximation and distributed implementation of Markov chains.

A. Log-Sum-Exp Approximation

First, the maximum broadcast rate can be approximated by a log-sum-exp function as follows:

$$\max_{f \in \mathcal{F}} x_f \approx \frac{1}{\beta} \log \left[\sum_{f \in \mathcal{F}} \exp(\beta x_f) \right], \quad (21)$$

where β is a positive constant. Let $|\mathcal{F}|$ denote the size of the set \mathcal{F} , then the approximation accuracy is known as follows [20]:

$$0 \leq \frac{1}{\beta} \log \left[\sum_{f \in \mathcal{F}} \exp(\beta x_f) \right] - \max_{f \in \mathcal{F}} x_f \leq \frac{1}{\beta} \log |\mathcal{F}|. \quad (22)$$

As β approaches infinity, the approximation gap approaches zero. As discussed in [20], however, usually β should not take too large values as there are practical constraints or convergence rate concerns in the algorithm design afterwards.

To better understand the log-sum-exp approximation, we associate with each configuration $f \in \mathcal{F}$ a probability p_f . Consider the following problem

$$\text{MRC - EQ} : \max_{p \geq 0} \sum_{f \in \mathcal{F}} p_f x_f \quad (23)$$

$$\text{s.t.} \quad \sum_{f \in \mathcal{F}} p_f = 1. \quad (24)$$

Its optimal value is $\max_{f \in \mathcal{F}} x_f$ and is obtained by setting the probability corresponding to one of the best configurations to be one and the rest probabilities to be zero. Hence, problem **MRC - EQ** is equivalent to the original problem **MRC**.

On the other hand, according to [20] we have the following observations.

Theorem 2 (cf. [20]): The optimal value of the following optimization problem

$$\text{MRC} - \beta : \max_{p \geq 0} \sum_{f \in \mathcal{F}} p_f x_f - \frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f \quad (25)$$

$$\text{s.t.} \quad \sum_{f \in \mathcal{F}} p_f = 1. \quad (26)$$

is given by $\frac{1}{\beta} \log \left[\sum_{f \in \mathcal{F}} \exp(\beta x_f) \right]$. The optimal solution of problem **MRC** - β is given by

$$p_f^*(\mathbf{x}) = \frac{\exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})}, \quad \forall f \in \mathcal{F}, \quad (27)$$

As such, by the log-sum-exp approximation in (21), we obtain an approximate version of the maximum broadcast rate problem **MRC**, off by an *entropy* term $-\frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f$. If we can time-share among different configurations according to the optimal solution $p_f^*(\mathbf{x})$ in (27), then we can solve the problem **MRC** approximately and obtain a close-to-optimal broadcast rate.

B. Markov Chain Guided Algorithm Design

We design a Markov chain with a state space being the set of all feasible peering configurations \mathcal{F} and has a stationary distribution as $p_f^*(\mathbf{x})$ in (27). We implement the Markov chain to guide the system to optimize the configuration. As the system hops among configurations, the Markov chain converges and the configurations are time-shared according to the desired distribution $p_f^*(\mathbf{x})$.

The key lies in designing such Markov chain that allows distributed implementation. Since $p_f^*(\mathbf{x})$ in (27) is product-form, it suffices to focus on designing time-reversible Markov chains [20].

Let $f, f' \in \mathcal{F}$ be two states of Markov chain, and denote $q_{f,f'}$ as the transition rate from state f to f' . We have two degrees of freedom in designing a time-reversible Markov chain:

- **The state space structure:** we can add or cut direct transitions between any two states, given that the state space remains connected and any two states are reachable from each other.
- **The transition rates:** we can explore various options in designing $q_{f,f'}$, given that the detailed balance equation is satisfied, i.e.,

$$p_f^*(\mathbf{x}) q_{f,f'} = p_{f'}^*(\mathbf{x}) q_{f',f}, \quad \forall f, f' \in \mathcal{F}. \quad (28)$$

Satisfying the above equations guarantees the designed Markov chain has the desired stationary distribution as in (27).

Recall that for a node $v \in V$, the set of its neighbors under configuration f is denoted by $N_{v,f}$. For the ease of explanation, we further define \mathcal{N}_f as the set of all the node-pairs under f , i.e., $\mathcal{N}_f \triangleq \{\{v, u\}, \forall v \in V, u \in N_{v,f}\}$. Note we do not differentiate node pairs $\{u, v\}$ and $\{v, u\}$. As an example, for the peering configuration f shown in Fig. 1(b), \mathcal{N}_f is given by $\{\{s, 1\}, \{s, 2\}, \{s, 4\}, \{1, 2\}, \{1, 4\}, \{2, 3\}, \{3, 4\}\}$.

In our Markov chain design, we first specify its state space structure as follows: we set the transition rate $q_{f,f'}$ to be zero, unless f and f' satisfy that

- $|\mathcal{N}_f \cup \mathcal{N}_{f'} - \mathcal{N}_f \cap \mathcal{N}_{f'}| = 2$, i.e., f and f' differ by only two node pairs.
- there exists a node, denoted by v^* , so that $\mathcal{N}_f \cup \mathcal{N}_{f'} - \mathcal{N}_f \cap \mathcal{N}_{f'} \subseteq \{\{v^*, u\}, \forall u \in N_{v^*}\}$. That is, these two node pairs share a common node v^* .

In other words, we only allow direct transitions between two configurations if such transitions correspond to a single node swapping an in-use neighbor with a not-in-use one.

Second, given the state space structure of Markov chain, we design the transition rates to favor distributed implementation while satisfying the detailed balance equation in (28).

One possible option is to set $q_{f,f'}$ to be $\exp^{-1}(\beta x_f)$. One way to implement this option is for every node to generate a timer according to its *measured* receiving rate and counts down accordingly. When the timer expires, the dedicated node performs the neighbor swapping and resets its timer. As simple as the implementation may sound, this option is expensive

to implement. Once the peering configuration changes, the system needs to notify all the nodes to measure the new receiving rate and reset their timers accordingly. It is not clear how to implement such system-wide notification in a low-overhead manner.

In this paper, we design $q_{f,f'}$ and $q_{f',f}$ as follows:

$$q_{f,f'} = \frac{1}{\exp(\tau)} \frac{\exp(\beta x_{f'})}{\exp(\beta x_f) + \exp(\beta x_{f'})} \quad (29)$$

and

$$q_{f',f} = \frac{1}{\exp(\tau)} \frac{\exp(\beta x_f)}{\exp(\beta x_f) + \exp(\beta x_{f'})}, \quad (30)$$

where τ is a constant. It is straightforward to verify that detailed balance equation is satisfied. As will be clear in the next subsection, our choices of transition rates do not require coordination or notification among peers in its implementation.

C. Distributed Implementation

One distributed implementation of our designed Markov chain is briefly described as follows.

- **Initialization:** Each peer $v \in V$ randomly selects B_v number of neighbors from its neighbor list N_v and builds connections with these selected neighbors.
- **Step 1:** Let f denote the current configuration. Each node $v \in V$ generates an exponentially distributed random number independently with mean $\frac{\exp(\tau)}{(|N_v| - |N_{v,f}|) |N_{v,f}|}$, and counts down according to this number.
- **Step 2:** When the count-down expires, node v measures its current receiving rate as an estimate of the broadcast rate x_f . Then node v randomly swapping one in-use neighbor in $N_{v,f}$ with a non-in-use one in N_v . Under the new peering configuration f' , node v measures its receiving rate as an estimate of $x_{f'}$. With the estimates of x_f and $x_{f'}$, peer v stays in the new configuration f' with probability $\frac{\exp(\beta x_{f'})}{\exp(\beta x_f) + \exp(\beta x_{f'})}$, and switches back to f with probability $1 - \frac{\exp(\beta x_{f'})}{\exp(\beta x_f) + \exp(\beta x_{f'})}$. Node v then repeats **Step 1**.

It is straightforward to summarize the above implementation into a distributed algorithm that runs on individual nodes and utilizes only the measurement from their one-hop neighbors. The correctness of the implementation is shown as follows:

Proposition 1: The implementation in fact realizes a time-reversible Markov chain with stationary distribution in (27).

Remarks: a) In **Step 1**, the generation of count-down timers does not depend on the receiving rate, thus the system does not need to notify the nodes about changes of peering configurations. b) With the above implementation, the system hops towards configurations with better broadcast rate probabilistically. For example, if $x_{f'} > x_f$, then the system will be more likely to stay in configuration f' than in f , and vice versa. c) With large values of β , the system hops towards better configurations more greedily. However, this may as well lead to the system getting trapped in locally optimal configurations. Hence there is a trade-off to consider when

setting the value of β . Moreover, the value of β also affects the convergence rate of the time-reversible Markov chain to the desired stationary distribution. It is worth future investigation to further understand the impact of β .

V. CONVERGENCE PROPERTIES OF OVERALL SOLUTION

We have designed the distributed broadcasting algorithm in Section III and the Markov chain guided topology hopping algorithm in Section IV. Both algorithms are simple to implement, run on each individual node, and only require nodes to exchange information with their neighbors.

If the broadcasting algorithm converges instantaneously, i.e., time-scale separation assumption holds, then we can obtain the accurate value of x_f for any configuration $f \in \mathcal{F}$. Transiting based on the accurate x_f , the designed Markov chain will converge to the desired stationary distribution in (27). Hence by operating these two algorithms in tandem, we obtain a close-to-optimal broadcast rate under arbitrary node degree bounds, and over arbitrary overlay graph. The optimality gap is characterized in (22).

In practice, however, it is possible to obtain only an inaccurate measurement or estimate of x_f . These inaccuracies root in two sources. One is the noisy measurements of the maximum broadcast rates given the configuration. The other is the fast state transition of Markov chain, i.e., the Markov chain transits before the underlying broadcasting algorithm converges and thus it transits based on inaccurate observations of the broadcast rates.

Consequently, the topology hopping Markov chain may *not* converge to the desired stationary distribution $p_f^*(\mathbf{x})$. This observation motivates our following study on the convergence of Markov chain in the presence of inaccurate transition rates.

For each configuration $f \in \mathcal{F}$ with broadcast rate x_f , we assume its corresponding inaccurate observed rate belongs to the bounded region $[-\Delta_f, \Delta_f]$. Δ_f is the inaccuracy bound and can be different for different f .

For easy explanation of our approach, we further assume the observed broadcast rate for configuration f only takes one of the following $2n_f + 1$ discrete values:

$$\left[x_f - \Delta_f, \dots, x_f - \frac{1}{n_f} \Delta_f, x_f, x_f + \frac{1}{n_f} \Delta_f, \dots, x_f + \Delta_f \right],$$

where n_f is a positive constant. Further, with probability $\eta_{j,f}$, the observed broadcast rate takes value $x_f + \frac{j}{n_f} \Delta_f$, $\forall j \in \{-n_f, \dots, n_f\}$ and $\sum_{j=-n_f}^{n_f} \eta_{j,f} = 1$.

With the inaccurate observed broadcast rates, the topology hopping behaves as follows. Suppose the current configuration is f and the observed broadcast rate is $x_f + \frac{j}{n_f} \Delta_f$, where $j \in \{-n_f, \dots, n_f\}$. After some count-down process, the system hops to a new configuration f' and probes its broadcast rate. In configuration f' , the broadcast rate is observed as $x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}$, $j' \in \{-n_{f'}, \dots, n_{f'}\}$. The system stays in the new configuration f' with probability

$$\frac{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}))}{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})) + \exp(\beta(x_f + \frac{j}{n_f} \Delta_f))},$$

and switches back to configuration f with probability

$$1 - \frac{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}))}{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})) + \exp(\beta(x_f + \frac{j}{n_f} \Delta_f))}.$$

By arguments similar to the proof of Proposition 1 in [21], the transition rate from configuration f with broadcast rate $x_f + \frac{j}{n_f} \Delta_f$ to configuration f' with broadcast rate $x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}$ is given by

$$\frac{\eta_{j',f'}}{\exp(\tau)} \cdot \frac{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}))}{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})) + \exp(\beta(x_f + \frac{j}{n_f} \Delta_f))}. \quad (31)$$

We construct a Markov chain to capture and study the above topology hopping behavior. In this Markov chain, a state is associated with a configuration and an observed broadcast rate. Given any configuration $f \in \mathcal{F}$ and its corresponding x_f , there are $2n_f + 1$ states in the extended Markov chain: $(f, x_f + \frac{j}{n_f} \Delta_f)$, $j \in \{-n_f, \dots, n_f\}$. Further, Given direct transitions between configuration f and f' in the original topology hopping Markov chain, there are direct transitions between states $(f, x_f + \frac{j}{n_f} \Delta_f)$ and $(f', x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})$ ($\forall j \in \{-n_f, \dots, n_f\}, j' \in \{-n_{f'}, \dots, n_{f'}\}$) in the corresponding new Markov chain. The corresponding transition rates are shown as follows:

$$\begin{aligned} & q_{(f, x_f + \frac{j}{n_f} \Delta_f), (f', x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})} \\ &= \frac{\eta_{j',f'}}{\exp(\tau)} \cdot \frac{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}))}{\exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'})) + \exp(\beta(x_f + \frac{j}{n_f} \Delta_f))} \end{aligned} \quad (32)$$

and

$$\begin{aligned} & q_{(f', x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}), (f, x_f + \frac{j}{n_f} \Delta_f)} \\ &= \frac{\eta_{j,f}}{\exp(\tau)} \cdot \frac{\exp(\beta(x_f + \frac{j}{n_f} \Delta_f))}{\exp(\beta(x_f + \frac{j}{n_f} \Delta_f)) + \exp(\beta(x_{f'} + \frac{j'}{n_{f'}} \Delta_{f'}))}, \end{aligned} \quad (33)$$

where $\sum_{j=-n_f}^{n_f} \eta_{j,f} = 1$ and $\sum_{j'=-n_{f'}}^{n_{f'}} \eta_{j',f'} = 1$. This new Markov chain can be thought as an extended version of the original topology hopping Markov chain. As an example, an extended Markov chain is shown and explained in Fig. 2.

The extended Markov chain has a unique stationary distribution since it is irreducible and only has a finite number of states. We can study the impact of inaccurate broadcast rates by comparing the stationary configuration distribution of the new Markov chain and that of the original topology hopping Markov chain.

We denote the stationary distribution of the *states* in the new Markov chain by

$$\tilde{\mathbf{p}} \triangleq [\tilde{p}_{f, x_f + \frac{j}{n_f} \Delta_f}, j \in \{-n_f, \dots, n_f\}, f \in \mathcal{F}] \quad (34)$$

We also denote $\bar{\mathbf{p}} : [\bar{p}_f(\mathbf{x}), f \in \mathcal{F}]$ as the stationary distribution of the *configurations* in the extended Markov chain. Given a

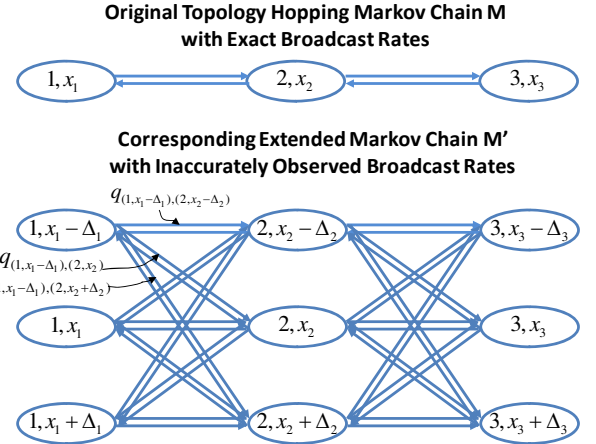


Fig. 2. An example of the original three-state topology hopping Markov chain and the extended Markov chain. M is the original topology hopping Markov chain with accurate broadcast rates. M' is the corresponding extended Markov chain with inaccurate broadcast rate observations. For each configuration $f \in \{1, 2, 3\}$, the observed broadcast rate takes values $x_f - \Delta_f$, x_f , $x_f + \Delta_f$ with probability $\eta_{-1,f}$, $\eta_{0,f}$ and $\eta_{1,f}$ respectively. The transition rates are assigned according to (32) and (33).

configuration $f \in \mathcal{F}$, there are $2n_f + 1$ states associated with f in the extended Markov chain. We have

$$\bar{p}_f(\mathbf{x}) = \sum_{j \in \{-n_f, \dots, n_f\}} \tilde{p}_{f, x_f + \frac{j}{n_f} \Delta_f}, \forall f \in \mathcal{F} \quad (35)$$

Recall that the stationary distribution of the configurations for the original topology hopping Markov chain is $\mathbf{p}^* : [p_f^*(\mathbf{x}), f \in \mathcal{F}]$. We use the total variance distance [30] to quantify the difference between \mathbf{p}^* and $\bar{\mathbf{p}}$, as

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \triangleq \frac{1}{2} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \quad (36)$$

We have the following result:

Theorem 3: Let $\Delta_{\max} = \max_{f \in \mathcal{F}} \Delta_f$, and $x_{\max} = \max_{f \in \mathcal{F}} x_f$. The $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ are bounded as follows:

$$0 \leq d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \leq 1 - \exp(-2\beta\Delta_{\max}). \quad (37)$$

Further, the optimality gap in broadcast rates $|\mathbf{p}^* \mathbf{x}^T - \bar{\mathbf{p}} \mathbf{x}^T|$ is bounded as below:

$$|\mathbf{p}^* \mathbf{x}^T - \bar{\mathbf{p}} \mathbf{x}^T| \leq 2x_{\max}(1 - \exp(-2\beta\Delta_{\max})). \quad (38)$$

Remarks: a) The upper bound on $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ shown in (37) is general, as it is independent of the number of configurations $|\mathcal{F}|$, the values of n_f , and the distributions of inaccurate observed rates $\eta_{j,f}$ ($-n_f \leq j \leq n_f, f \in \mathcal{F}$). b) The upper bound on $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ shown in (37) decreases exponentially with the worst inaccuracy bound Δ_{\max} decreasing. c) It would be interesting to explore a tighter upper bound on $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ than the one in (37).

VI. PERFORMANCE EVALUATION

We implement our proposed solution using Python and carry out simulations to evaluate the performance of our solution.

TABLE II
PEER UPLOAD CAPACITY DISTRIBUTION

Upload Capacity (kbps)	64	128	256	384	768
Fraction (%)	2.8	14.3	4.3	23.3	55.3

A. Settings

In our simulations, time is chopped into slots of equal length, and we adopt three different settings. In Setting I, we set the total number of nodes to be 100, and assign the node upload capacities randomly according to the distribution in Table II, which is obtained from the uplink bandwidth statistics of Internet hosts [31]. We set the source's upload capacity to be 768 kbps; with this upload capacity, source is not the broadcast bottleneck [1], [3].

Setting II is the same as Setting I, except we set the total number of nodes to be 10.

In Setting III, there are 4 different peering configurations as shown in Fig. 3. Every node has a unit capacity. Under configuration f_1 and f_2 the maximum broadcast rate is 1, and under configuration f_3 and f_4 the maximum broadcast rate is 0.5.

When running our network coding based broadcasting algorithm, we set the updating step size of z and λ to be 0.1 and 0.005 respectively. These parameters are empirically chosen to obtain smooth algorithm updating and small errors.

In our simulations, we assign node degree bounds in the following two ways. The first is to set identical bound on each node's node degree. The second is to set degree bound proportional to the node's upload capacity. This is based on the empirical observations that nodes with high upload capacities usually have more system resource (e.g., memory and CPU power) than nodes with low upload capacities. With more system resource, nodes can maintain more concurrent connections, thus have larger node degree bounds. In our second degree bounds assignment, nodes set their node degree bounds proportional to the ratio between their upload capacities and 64 kbps. In particular, nodes with 64 kbps have a degree bound of 1, and nodes with 128 kbps have a degree bound of 2, etc.

We carry out two sets of simulations. First, we evaluate the performance of our distributed broadcasting algorithm under Setting I and II. Second, we evaluate the overall performance when we combine the topology hopping algorithm and the broadcasting algorithm under Setting I and III. In these two sets of simulations, we also compare the performance under the two degree bounds assignments explained in the previous paragraph.

B. Evaluation of the Proposed Broadcasting Algorithm

In this simulation, we evaluate our distributed broadcasting algorithm proposed in Section III. We randomly choose a sub-graph that satisfies the node degree bounds constraints, and run our algorithm over it. We evaluate three aspects of the proposed algorithm: 1) does it converge to optimal broadcast rate as expected from theoretical analysis? 2) How fast does it converge? 3) How would different values of degree

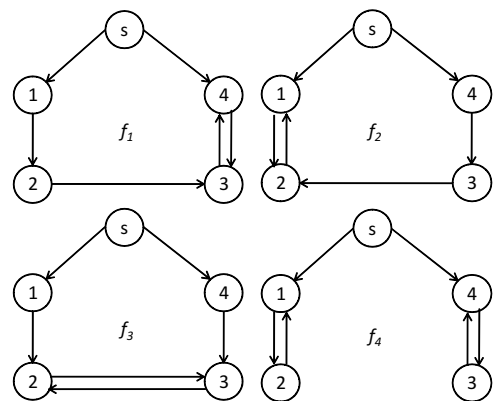


Fig. 3. Peering configurations under Setting III. For the ease of illustration, we only allow node 2 to switch neighbors between nodes 1 and 3 and node 3 to switch between nodes 2 and 4. The rest nodes keep their neighbors fixed.

bounds affect the maximum broadcast rate? The results are summarized in Fig. 4.

From Fig. 4(a) and Fig. 4(b), we see that our broadcasting algorithm converges. It converges faster in the small size network as shown in Fig. 4(a) than in the large size network as shown in Fig. 4(b). From Fig. 4(d), we also see the converged rate when the node degree bound is 10 is very close to a theoretical upper bound – the optimal broadcast rate under no degree bounds computed according to [1] [17] [3]. This suggests that our algorithm converges to the optimal broadcast rate.

Under different degree bounds, the optimal broadcast rate varies. Fig. 4(d) shows that the optimal broadcast rate increases when we increase the node degree bounds. We plot the CDF of peer receiving rates (after the broadcasting algorithm converges) for the case where degree bound is 3, 10, and proportional to the peer's upload capacity. It's seen that when the bound is 10, the obtained rate is close to the full-mesh rate, which suggests that we do not need a large degree bound to achieve close to the full-mesh rate. The obtained rate is also close to the full-mesh rate when degree bound is proportional to the peer's upload capacity.

C. Evaluation of the Overall Solution

Our overall solution combines the topology hopping algorithm and the broadcasting algorithm to achieve the near-optimal broadcast rate under arbitrary node degree bound and over arbitrary overlay graph. To evaluate its performance, we generate a sub-graph randomly, run our algorithms on every node, and evaluate the achieved broadcast rate.

The topology hopping algorithm runs on top of the broadcasting algorithm. Under given topology, the broadcasting algorithm achieves the optimal broadcast rate. Nodes swap neighbors based on their observed receiving rate, thus changing the topology from time to time. In the simulation, we run the broadcasting algorithm long enough so that it converges before the topology transits according to the Markov chain. This way, the overall algorithm converges to the close-to-optimal broadcast rate.

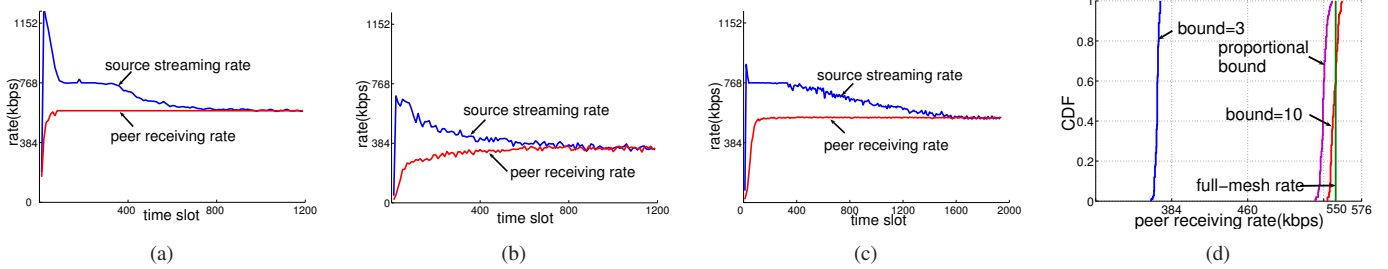


Fig. 4. Broadcasting algorithm evaluations. a) The source broadcast rate and average peer receiving rate under Setting II when degree bound is set to 3; b) The source broadcast rate and average peer receiving rate under Setting I when degree bound is set to 3; c) The source broadcast rate and average peer receiving rate under Setting I when peer degree bound is proportional to its upload capacity. d) This figure shows the impact of degree bound on the peer receiving rate under Setting I. The full-mesh rate is the maximum broadcast rate when the node degrees are unbounded [1].

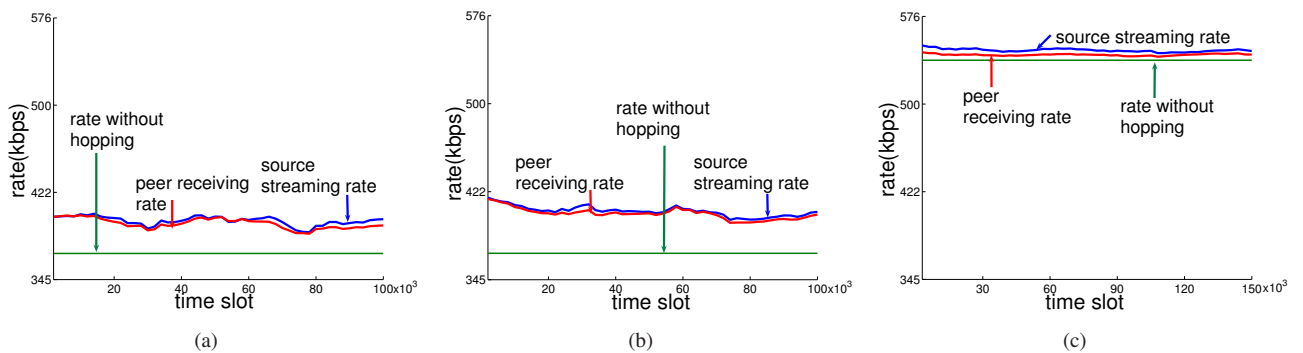


Fig. 5. Evaluation of our overall solution by combining the topology hopping algorithm and the broadcasting algorithm. a) The source broadcast rate and average peer receiving rate when the node degree bound is 3 and β is 5; b) The source broadcast rate and average peer receiving rate when the node degree bound is 3 and β is 20; c) The source broadcast rate and average peer receiving rate when peer degree bound is proportional to its upload capacity and β is 5.

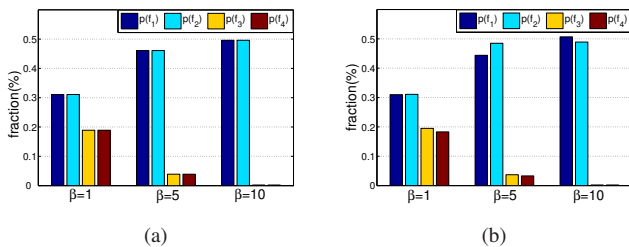


Fig. 6. a) Optimal configuration distribution for different values of β under Setting III; b) Configuration distribution obtained by our algorithm for different values of β under Setting III.

Our first observation is that the overall scheme converges to the solution that theory predicts. We carry out simulations under Setting III. Under this setting the optimal broadcast rate is 1. The optimal configuration solution to problem $\text{MRC} - \beta$ is calculated and shown in Fig. 6(a) for different values of β . We run the overall scheme for this specific case and show the empirical configuration distribution in Fig. 6(b). Comparing the distributions in Fig. 6(a) and Fig. 6(b), we can see that the distribution obtained by our overall solution is very close to the optimal one. We also calculate the achieved broadcast rate under different values of β . For $\beta = 1, 5$ and 10 , the broadcast rate is $0.81, 0.964$, and 0.998 respectively. We see that with large β , the achieved broadcast rate is close to the optimal

value 1, as predicted by our analysis in Section IV.

Next, we evaluate the overall solution under Setting I. In Fig. 5(a) and Fig. 5(b), the broadcast rates obtained are 398 kbps and 405 kbps respectively. They are about 10% higher than the broadcast rate 368 kbps achieved by running the broadcasting algorithm over a randomly chosen topology, as shown in Fig. 4(b). This demonstrates the advantage of performing topology hopping to optimize the configuration, as compared to only randomly choosing topology.

By setting node degree bounds proportional to peers' upload capacity, nodes with higher upload capacity maintain more connections. From Fig. 5(c), we observe that this flexibility offers a broadcast rate of 545 kbps, which is close to the optimal broadcast rate achieved under no node degree bounds. This illustrates the benefit of allowing nodes degree bounds to be proportional to their upload capacities.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a distributed solution to achieve a near-optimal broadcast rate under arbitrary node degree bounds, and over arbitrary overlay graph. Our solution is distributed and consists of two algorithms that can be of independent interests. The first is a distributed broadcasting algorithm that optimizes the broadcast rate given a P2P topology. It is derived from a network coding based problem formulation and utilizes back-pressure arguments. It can be

considered as the extension of the algorithm in [25] from link-capacity-limited underlay networks to node-capacity-limited overlay networks. The second algorithm is a Markov chain guided hopping algorithm that optimizes the topology, inspired by the Markov Approximation framework introduced in [20].

Assuming the underlying broadcasting algorithm converges instantaneously, the topology hopping algorithm converges to the optimal configuration distribution. When the broadcasting algorithm does not converge fast enough, the topology hopping Markov chain transits based on inaccurate observations of the maximum broadcast rates associated with the configurations. We show that the topology hopping algorithm still converges, but to a sub-optimal configuration distribution. We characterize an upper bound on the total variance distance between the optimal and sub-optimal configuration distributions, as well as an upper bound on the gap between the achieved and the optimal broadcast rates. We show that both bounds decrease exponentially as the bound on inaccuracy decreases.

Using uplink bandwidth statistics of Internet hosts, our simulations validate the effectiveness of the proposed solutions, and demonstrate the advantage of allowing node degree bounds to scale linearly with their upload capacities.

Two interesting future directions are as follows. First, similar to other network-coding based resource allocation algorithms, our broadcasting algorithm requires every node to maintain one virtual queue for every receiver. This causes scalability concerns in large P2P systems. It is interesting to investigate how to reduce the queue management overhead of the broadcasting algorithm.

Second, the convergence rate of our solution is determined by the mixing time of the topology-hopping Markov chain, which can be substantial for large P2P systems. It is thus of great interest to explore the design of topology-hopping Markov chains that mix fast and at the same time allows distributed implementation.

ACKNOWLEDGEMENT

The authors would like to thank Libin Jiang for the helpful discussions. This work was partially supported by the General Research Fund grants (Project No. 411008, 411209, 411010) and an Area of Excellence Grant (Project No. AoE/E-02/08), all established under the University Grant Committee of the Hong Kong SAR, China. This work was also partially supported by two gift grants from Microsoft and Cisco.

REFERENCES

- [1] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: an efficient mechanism for content distribution in a p2p network," in *Proceedings of Acm Sigcomm Asia Workshop*, Beijing, China, Apr. 2005.
- [2] L. Massoulié, A. Twigg, G. Gkantsidis, and P. Rodriguez, "Randomized Decentralized Broadcasting Algorithms," in *Proc. IEEE INFOCOM*, Anchorage, AL, May 2007.
- [3] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for p2p streaming systems," in *Proc. IEEE INFOCOM*, Anchorage, AL, USA, 2007.
- [4] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal resource allocation in overlay multicast," *IEEE Transaction on Parallel and Distributed Systems*, August 2006.

- [5] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. A. Chou, "Streaming capacity for peer-to-peer networks with topology constraints," *submitted to IEEE Trans. on Information Theory (in revision)*, 2009.
- [6] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments," in *Proc. ACM SOSP*, New York, NY, October 2003.
- [7] V. Padmanabhan and K. Sripanidkulchai, "The case for cooperative networking," in *Proceedings of IPTPS*, Cambridge, MA, March 2002.
- [8] D. A. Tran, K. Hua, and T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *Proceedings of IEEE Infocom*, 2003.
- [9] N. Magharei and R. Rejaie, "PRIME: Peer-to-peer receiver-driven mesh-based streaming," *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1052–1065, 2009.
- [10] S. Liu, M. Chen, S. Sengupta, M. Chiang, J. Li, and P. A. Chou, "Peer-to-peer streaming capacity under node degree bound," in *Proc. of IEEE ICDCS*, Genoa, Italy, June 2010.
- [11] C. Feng, B. Li, and B. Li, "Understanding the performance gap between pull-based mesh streaming protocols and fundamental limits," in *Proc. of IEEE INFOCOM*, 2009.
- [12] L. Abeni, C. Kiraly, and R. L. Cigno, "On the optimal scheduling of streaming applications in unstructured meshes," in *IFIP Networking*, 2009.
- [13] M. Wang and B. Li, "R²: Random Push with Random Network Coding in Live Peer-to-Peer Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, p. 1655, 2007.
- [14] X. Zhang, J. Liu, B. Li, and T. Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," in *proc. of IEEE Infocom*, vol. 3, 2005, pp. 13–17.
- [15] M. Chen, M. Chiang, P. A. Chou, J. Li, S. Liu, and S. Sengupta, "Peer-to-peer streaming capacity: Survey and recent results," in *Proc. of Allerton Conference*, Urbana, Illinois, USA, September 2009.
- [16] D. M. Chiu, R. W. Yeung, J. Huang, and B. Fan, "Can network coding help in p2p networks?" in *Proc. of IEEE NetCod*, Boston, MA, USA, April 2006.
- [17] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. Chou, "Utility maximization in peer-to-peer systems," in *Proc. ACM SIGMETRICS*, Annapolis, MD, June 2008.
- [18] Pplive - internet peer-to-peer video streaming. [Online]. Available: <http://www.pplive.com>
- [19] Uusee. [Online]. Available: <http://www.uusee.com>
- [20] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," in *Proc. of IEEE INFOCOM*, San Diego, CA, USA, 2010.
- [21] S. Zhang, Z. Shao, and M. Chen, "Optimal distributed p2p streaming under node degree bounds," The Chinese University of Hong Kong, Hong Kong, Tech. Rep., 2010. [Online]. Available: http://www.ie.cuhk.edu.hk/~mhchen/papers/dist_p2p_streaming.pdf
- [22] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, no. 4, pp. 1204–1216, Jul. 2000.
- [23] Z. Liu, C. Wu, B. Li, and S. Zhao, "Uusee: Large-scale operational on-demand streaming with random network coding," in *Proc. IEEE INFOCOM*, San Diego, CA, US, Mar. 2010.
- [24] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.
- [25] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *Information Theory, IEEE Transactions on*, vol. 55, no. 2, pp. 797–815, feb. 2009.
- [26] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conference*, 2003.
- [27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [28] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhäuser, 2004.
- [29] L. Georgiadis, M. Neely, and L. Tassiulas, *Resource allocation and cross layer control in wireless networks*. Now Pub, 2006.
- [30] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of Markov chains," *The Annals of Applied Probability*, pp. 36–61, 1991.
- [31] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" in *ACM SIGCOMM*, Kyoto, Japan, August 2007.