Asymptotic Optimality of Randomized Peer-to-Peer Broadcast with Network Coding

Di Niu, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto

***Abstract*—We consider the problem of distributing k blocks from a source to N nodes in a peer-to-peer (P2P) network with both node upload and download capacity constraints. As k scales up, we prove for homogeneous networks that if network coding is allowed, randomly matching senders and receivers in each time slot asymptotically achieves the maximum downloading rate at each node. For heterogeneous networks with network coding allowed, we show that a fair and optimal downloading rate at each node can be asymptotically approached, if in each time slot, each node randomly allocates its upload bandwidth to its receivers that have available download bandwidth. We also give a performance lower bound of the above randomized coded dissemination when both k and N scale under certain conditions. These results demonstrate that with network coding, simple randomized receiver selection and rate allocation suffice to achieve P2P broadcast capacity, forming a theoretical foundation for mesh-based P2P networks with network coding.**

I. Introduction

We consider the problem of broadcasting k data blocks from a single source to N nodes in a peer-to-peer (P2P) network, described by a directed graph $G = (V, E)$, where nodes represent end hosts on the Internet, and edges represent TCP or UDP connections between the end hosts. Such a P2P broadcast problem, as an abstract model for P2P content distribution and media streaming systems, has attracted much research attention in recent years. It is typical to assume that node capacity constraints are bottlenecks in P2P networks, rather than the core of the Internet. As a result, each node has to make the dual decision of both rate allocation (among its neighboring nodes) and link scheduling, with the goal of maximizing its download rate or minimizing its download finish time.

It has been shown [1]–[3] that when G is a complete graph, the optimal broadcast rate can be achieved by packing spanning trees. Distributed solutions have also been proposed based on node buffer state exchanges [4] or primal-dual algorithms based on queuing delays [5]. However, these algorithms require non-trivial signaling overhead, and their convergence and success depends critically on the frequency and accuracy of control message updates.

Randomized broadcast algorithms, also known as gossiping, let each node transmit to random neighbors, and offer a promising decentralized solution as an alternative, due to their simplicity and superior robustness with the presence of node dynamics. However, randomized algorithms are often known to be best-effort: even with coding allowed, communicating with random neighbors is only shown to be order-optimal [6]. On the other hand, it is well known that given a rate allocation schedule, network coding is superior to any block selection scheme [7]. However, it is unclear to what extent coding can simplify the design of rate allocation or receiver selection schemes, while still achieving optimal download rates.

In this paper, unlike most previous works [1]–[6], [8] that assume node uplinks are the only bottlenecks, we consider both node uplink and downlink capacity constraints. With analytical rigor, we prove that as the number of data blocks k scales up in homogeneous networks, if each node transmits a random linear combination of blocks it has obtained, the optimal download rates at the nodes can be achieved by randomly matching the upload and download ports of all the nodes in each time slot. In heterogeneous networks with coding allowed, if each node randomly allocates its upload bandwidth to other nodes with available download bandwidth, the optimal download rates are approximately achieved as k scales. When $k = \Omega(N^2)$ and N scales, we show that the above simple randomized broadcast with coding is within e^{-1} of the optimal solution.

In our analysis, we convert the original algebraic coding problem into flow arguments by modelling the block transmission on a time-expanded trellis, and carefully bound large deviations using probabilistic arguments. Our theoretical results imply that when coding is allowed, the optimal broadcast performance can be approached even with simple randomized rate allocation, which is robust and amenable to a decentralized implementation in dynamic networks. The use of network coding completely eliminates sophisticated link scheduling and rate allocation, as well as node buffer state exchanges of any form.

The remainder of this paper is organized as follows. Sec. II introduces our system model. Sec. III discusses the dissemination capacity of a P2P network and reviews prior work and existing methods that approach such capacity. We present our main results and their implications in Sec. IV. We prove the optimality of randomized broadcast with coding as k scales in Sec. V, and provide performance bounds when both k and N scale in Sec. VI. We present simulation results in Sec. VII and conclude the paper in Sec. VIII.

II. System Model

We model a P2P network as a directed graph $G = (V, E)$, with nodes V representing end hosts and edges E representing connections between end hosts. There is a single source $s \in V$

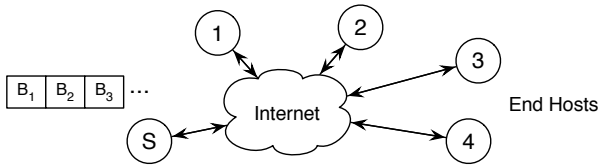

Fig. 1. Broadcasting multiple data blocks in P2P networks, with both uplink and downlink bandwidth constraints at each end host.

that wishes to broadcast its content of size B to $N = |V| - 1$ receiver nodes $\{1, 2, \ldots, N\}$. An edge $e = (u, v)$ exists in G if node u is allowed to transmit to node v via a UDP or TCP connection. Similar to P2P topology models in existing literature [1]–[6], [8], [9], we assume that a node is able to establish an overlay connection with any other node, so that G forms a full mesh.

In the majority of residential broadband connections, bandwidth bottlenecks typically lie at the edge of the access networks rather than in the core of the Internet, as shown in Fig. 1. The upload and download rates of each node $v \in V$ are bounded by its upload capacity U_v and download capacity D_v, respectively. Furthermore, users usually have asymmetric download and upload capacity. We thus assume $D_v \geq U_v$ for all $v \in V \backslash \{s\}$. We further assume $U_s \geq U_v$ for all $v \in V \backslash \{s\}$. Note that unlike previous works [1]–[6], [8], [9] that assume the download capacity of each node is unbounded, in this paper, we consider both node upload and download bandwidth constraints.

In reality, content from the source s is not transmitted as a commodity flow. Instead, it is composed of countably many *blocks*. During a block transmission between two nodes, the block is either completely downloaded by the receiver or not downloaded at all. Assume the content is divided into k blocks $\{\mathbf{B}_1, \ldots, \mathbf{B}_k\}$ for distribution, each of which has a unit size. Under such a packetized workload, it is reasonable to assume that time is measured in slots [6]–[10], the length of each slot being the time it takes a node to upload one block at the unit rate. In this case, the upload capacity U_v and download capacity D_v of node $v \in V$ are measured by how many blocks the node can upload and download in each time slot.

A transmission strategy at each node makes decisions on how receivers in each slot are chosen, how rates are allocated to each receiver, and which blocks are to be transmitted. Such a strategy could be either centralized or decentralized. For decentralized strategies, receiver selection and rate allocation can be either dependent or independent of block selection. Block selection algorithms can be classified as follows:

Block Scheduling. During a block transmission, the sender has to select a block from all the original blocks it has obtained so far to transmit to the receiver.

Network Coding. During a block transmission from the source in time slot t, the source s sends out a coded block $\mathbf{B}_c(t)$ that is a linear combination of the original blocks, *i.e.,* $\mathbf{B}_c(t) = \sum_{i=1}^{k} c_i(t)\mathbf{B}_i$, where $c_i(t)$ is a random coefficient chosen from the Galois field $GF(2^q)$ for each t and each $i \in \{1, \ldots, k\}$. During a block transmission from a node $v \in V \backslash \{s\}$ in time slot t, node v sends out a coded block $\mathbf{B}_c(t)$ that is a linear combination of the blocks it has received so far. Specifically, if v has received $l(t)$ coded blocks $\{\mathbf{B}_c^i | i = 1, \ldots, l(t)\}$ by time t, then it will send the block $\mathbf{B}_c(t) = \sum_{i=1}^{l(t)} c_i(t)\mathbf{B}_c^i$, where $c_i(t)$ is a random coefficient chosen from the Galois field GF(2^q) for each t and each $i \in \{1, \ldots, l(t)\}$. Each node has to receive k linearly independent coded blocks in order to decode the original content.

An ideal transmission strategy should be decentralized, requiring no state exchange among nodes, resilient to peer joins and departures, and easy to implement.

III. P2P Dissemination Capacity and Prior Work

The dissemination ability of a transmission strategy can be evaluated by the achievable download rates and download finish times. Let $T_v(k)$ be the time it takes node v to finish downloading all k blocks. If $k \to \infty$, *i.e.*, the content is infinitely large, we can define the asymptotic download rate of each node as

$$r_i := \lim_{k \to \infty} \frac{k}{T_i(k)}, \quad i = 1, 2, \ldots, N, \tag{1}$$

Clearly, the download rate of each node i is limited by its download capacity D_i and the maximum rate that the source sends blocks out, U_s. Furthermore, the aggregate download rate of all the receivers cannot exceed the aggregate upload capacity of the network. Thus, an achievable rate vector $\boldsymbol{r} = (r_1, \ldots, r_N)$ must satisfy

$$r_i \leq \min\{U_s, D_i\}, \quad i = 1, 2, \ldots, N, \tag{2}$$

$$\sum_{i=1}^{N} r_i \leq \sum_{i=1}^{N} U_i + U_s. \tag{3}$$

We define the optimal rate vector as follows:

Definition 1 (Optimality): A rate vector $\boldsymbol{r}^* = (r_1^*, \ldots, r_N^*)$ achieved under a certain transmission strategy is optimal if increasing either one of r_i^*, $i = 1, \ldots, N$ will result in a new rate vector that violates (2)-(3).

For the system to scale as N grows, there should be a balance between the download rate at each node and its contribution. We thus define a fairness measure as follows:

Definition 2 (Fairness): A rate vector $\boldsymbol{r} = (r_1, \ldots, r_N)$ achieved under a certain transmission strategy is fair if $r_i/U_i = r_j/U_j$, for all $i, j \in \{1, \ldots, N\}$.

Although (2)-(3) form a necessary condition for the rate vector $\boldsymbol{r}$ to be achievable, it is not a sufficient condition. Under the special case of unbounded download capacity and uniform downloading rates, *i.e.*, $D_i = \infty$ for all i, and $r_i = r$ for all i, (2)-(3) are simplified to the following:

$$r \leq \min\{U_s, \frac{U_s + \sum_{i=1}^{N} U_i}{N}\}. \tag{4}$$

Under the assumption that the content to be broadcast forms a flow, *i.e.*, the content is infinitely divisible, it is a well known result that (4) can be achieved with equality by packing spanning trees of depth at most two [1]–[3]. When blocks are sufficiently small, whatever rate vectors that can be achieved at

a flow level will be achieved under the time-slotted packetized workload with a vanishing error.

However, tree-packing algorithms are usually centralized. Their distributed implementation, *e.g.*, the primal-dual algorithm [5] for tree construction and rate allocation, is complicated, requiring non-negligible control message passing between nodes, and may not be stable or adapt well in the presence of node dynamics.

Randomized gossip [6], [8] provides an alternative solution with unique benefits over tree-based algorithms. In a gossip algorithm, each node selects random receivers to send blocks to in each time slot. Due to their simplicity, randomized gossip algorithms are easy to implement, extremely scalable and inherently resilient to node dynamics. Nonetheless, these algorithms are usually believed to be best-effort algorithms that do not guarantee the optimal performance.

Under the time-slotted model, when $U_v = 1$ for all $v \in V$, a well known bound of download finish time [8] is that, if each node has a sufficiently large download capacity, *i.e.*, $D_v = \infty, \forall v \in V$, the finish time of the last node $\max_v T_v(k)$, or the broadcast finish time, must satisfy

$$\max_v T_v(k) \geq k + \lceil \log_2 N \rceil, \tag{5}$$

because it takes at least k slots for the last block to emerge from the source, and a further $\lceil \log_2 N \rceil$ slots for that block to reach all users. It has been shown in [9], [10] that fully centralized block scheduling can achieve (5) with equality. When $N = O(k)$, under the centralized optimal strategy, for all $i \in \{1, \ldots, N\}$,

$$r_i^* = \lim_{k\to\infty} \frac{k}{T_i(k)} \geq \lim_{k\to\infty} \frac{k}{k + \lceil \log_2 N \rceil} = 1. \tag{6}$$

Since $r_i^* \leq U_s = 1$, we have $r_i^* = 1$, which also conforms to the optimality criterion in Definition 1.

However, no decentralized randomized gossip algorithms are known to achieve the optimal rate. It is shown in [8] that it takes $9(k + \log N)$ time slots to spread k blocks to N receivers based on a block scheduling protocol using both pushes and pulls, without buffer state exchanges between nodes. For network coding, it is shown in [6] that if each node holds one of the k blocks at the beginning, and in each time slot each node transmits a coded block to a random receiver using network coding, the download finish time of the last node is $ck + O(\sqrt{k}\log(k)\log(N))$, where c is a constant around $3 - 6$. Although these results imply order-optimal download rates, there still exists a performance gap between these algorithms and the optimal centralized strategy.

If buffer state exchanges between nodes are allowed, it is shown in [8] that the download finish time can be reduced to $N + \log(N)$, if $k = N$ and each node holds one distinct block to start with. This leads to an asymptotically optimal download rate of 1 at each node. With the requirement of node state reconciliation, a related result [4] shows that if each node u always transmits to a node v that maximizes the difference between the sets of blocks that u and v possess, the optimal broadcast rate can be achieved. However, it is theoretically shown [11] that a major obstacle of gossip protocols that require state exchanges is that their success critically depends on the accuracy and frequency of state updates.

In order to model realistic P2P networks more closely, in this paper, we consider the case where k blocks are only possessed by the source as an initial state. Different from previous works [1]–[6], [8], [9] that assume unbounded node download capacities, in this paper, we consider the general and more practical case with both node upload and download bandwidth constraints. We ask the question — is there a simple decentralized algorithm that requires almost no state exchanges between nodes, but can still achieve optimal download rates at the nodes?

In this paper, we advance the theoretical understanding of the benefit of network coding in P2P broadcast, by proving that if coding is allowed, simple randomized receiver selection and rate allocation, which requires no buffer state exchange, will achieve the optimal download rates in an asymptotic sense. We derive theoretical results in both homogeneous and heterogeneous networks as k and N scale.

IV. Main Results and Discussions

We define a *homogeneous network* as a network where $U_v = 1$ for all $v \in V$, and a *heterogeneous network* as a network with non-uniform node upload capacities. In all cases, our proposed algorithms apply to any finite D_v values such that $D_v \geq U_v$ for all $v \in V$.

For homogeneous networks, we define the following simple randomized receiver selection algorithms:

Random Receiver Selection. In each time slot, starting from an arbitrary node, following a random order, each node $u \in V$ randomly selects another node $v \in V$ that has not been chosen in this slot as its receiver.

Random Sender-Receiver Matching. In each time slot, the sender-receiver pairs form a random matching between all upload and download ports of the nodes such that no node is uploading to itself.

In the worst case of random receiver selection, the node that chooses last may end up with no choice but choosing itself as a receiver. Random sender-receiver matching is thus a stronger version of random receiver selection. It is easy to check that under both algorithms, a node can download at most 1 block in each time slot. Thus, these algorithms apply to any finite $D_v \geq U_v = 1$, for all $v \in V$.

For heterogeneous networks, recall that the upload or download capacity is a multiple of the unit bandwidth, which represents the ability to transmit one block in one time slot. We define the following simple randomized rate allocation scheme for the case of symmetric bandwidth, *i.e.*, $D_v = U_v$, for all $v \in V$. Without sacrificing performance, the case of asymmetric bandwidth can be converted to the symmetric case.

Random Rate Allocation. Assume symmetric bandwidth. In each time slot, starting from an arbitrary node, following a random order, each node $v \in V$ allocates every unit of its upload bandwidth to a random available download bandwidth unit of any node, until all its U_v upload bandwidth is allocated.

For the asymmetric case of $D_v > U_v$, we let $D'_v = U_v$ and only use the D'_v part of the download bandwidth of node v. Thus, we arrive at the same situation as in the symmetric case, and the random rate allocation can be applied accordingly.

As k scales while N is fixed, we have the following theorem for homogeneous networks.

Theorem 1: Assume the network is homogeneous with $U_v = 1 \leq D_v < \infty$, for all $v \in V$. If random sender-receiver matching is applied with random network coding, then with probability 1, the asymptotic download rate of node i is

$$r_i := \lim_{k\to\infty} \frac{k}{T_i(k)} = 1, \quad i = 1, 2, \ldots, N. \tag{7}$$

Furthermore, if random receiver selection is applied with random network coding, with probability 1,

$$r_i := \lim_{k\to\infty} \frac{k}{T_i(k)} > 1 - \frac{1}{N(N+1)}, \quad i = 1, 2, \ldots, N. \tag{8}$$

Apparently, in homogeneous networks, $r_i = 1$ for $i = 1, \ldots, N$ form an optimal rate vector by Definition 1, and both (7) and (8) give fair rate vectors by Definition 2.

There are several implications from this theorem. First, as compared to [6] that shows network coding based gossip is order-optimal in homogeneous networks with unbounded download capacity, Theorem 1 proves that if coding is allowed, simple randomized receiver selection can be absolutely optimal as k scales, even if both node upload and download capacities are constrained. Second, Theorem 1 implies that if network coding is allowed, a fair and optimal (7) or an approximately optimal (8) rate vector can be achieved using simple randomized receiver selection algorithms, whose decentralized implementation is much more feasible in practice than a tree-based approach. Randomized algorithms are also naturally resilient to node dynamics, with no need for structural updates or maintenance.

Furthermore, other known optimal decentralized algorithms based on block scheduling [4] require non-trivial buffer reconciliation. For example, the optimal scheduling algorithm in [4] requires each node u to transmit to a node v that maximizes the difference between the sets of blocks that u and v possess. The buffer reconciliation overhead grows rapidly as N scales. On the other hand, frequent and accurate buffer state exchanges are always hard to achieve, an important factor that leads to inefficiency in mesh-based P2P systems [11]. The use of network coding completely eliminates any buffer state exchanges or reconciliation, making the protocol simpler and more robust.

For heterogeneous networks, we can generalize Theorem 1 to have the following theorem:

Theorem 2: Assume the network is heterogeneous with $U_v \leq D_v < \infty$ for all $v \in V$. If random rate allocation is applied with random network coding, for $i = 1, \ldots, N$, with probability 1, the asymptotic download rate of node i is

$$r_i := \lim_{k\to\infty} \frac{k}{T_i(k)} = U_i\left(1 - \frac{U_i}{U_s + \sum_{i=1}^{N} U_i}\right). \tag{9}$$

While most prior work [4], [6], [8] focuses on homogeneous networks, Theorem 2 shows that for heterogeneous networks, with network coding, even simple randomized rate allocation independent of node buffer states can achieve fair and approximately optimal rates for sufficiently large N. In other words, node i finishes downloading all k blocks at time $T_i(k) \approx k/U_i$ as k scales. Since the total download rate of all the nodes cannot exceed the total upload rate, it is intuitively fair that each node enjoys an asymptotic download rate that is proportional to its upload contribution.

Finally, as both k and N scale, we have the following result for the expected time-average download rate:

Theorem 3: Assume the network is homogeneous with $U_v = 1 \leq D_v < \infty$, for all $v \in V$. Assume $k = \Omega(N^2)$ is sufficiently large. Let $B_v(t)$ be the number of linearly independent blocks that node v has received by time t. Let $t = cN^2$, c being a constant. If random receiver selection is applied with random network coding, we have

$$\lim_{t\to\infty} \frac{\mathbf{E}[B_v(t)]}{t} \geq 1 - \mathrm{e}^{-1}. \tag{10}$$

Theorem 3 implies that if $k = \Omega(N^2)$, as N scales, the expected download rate of each node in the long run is at least 63% of the optimal rate. In general, the use of network coding decouples block selection (encoding) and rate allocation into two completely independent processes, each of which can be realized using simple online randomized algorithms.

V. The Asymptotic Optimality

In this section, we develop the proof of Theorem 1 that shows the optimality of the proposed random receiver selection with coding in homogeneous networks as k scales while N is fixed. We also extend the results to heterogeneous networks and prove Theorem 2, which demonstrates the sufficiency of random rate allocation when coding is allowed.

A. Intuitions behind Theorem 1

We consider a trellis graph $G^* = (V^*, E^*)$ constructed from the nodes V, as shown in Fig. 2. For all $v \in V$, let $v_t \in V^*$ represent node v at time t. There is a directed edge of capacity 1 joining v_t and u_{t+1} if node v transmits a packet to node u at time t. To model the information accumulation at the nodes, for each node $v \in V$, we link v_t along the time line with edges of infinite capacity, denoted by the dashed lines. We have the following lemma given by Yeung [7]:

Lemma 1: Let $\text{maxflow}(v_t)$ be the value of the maximum flow from node s_0 to node $v_t \in V^*$. Given any receiver selection schedule, the number of *linearly independent* blocks that a node v can obtain by time t, $B_v(t)$, has the upper bound

$$B_v(t) \leq \min\{k, \text{maxflow}(v_t)\}. \tag{11}$$

Furthermore, (11) is achieved with equality by applying a random linear code at each node over a sufficiently large finite field [12].

Given k blocks for distribution, (11) follows directly from the celebrated max-flow bound for multicast in acyclic graphs

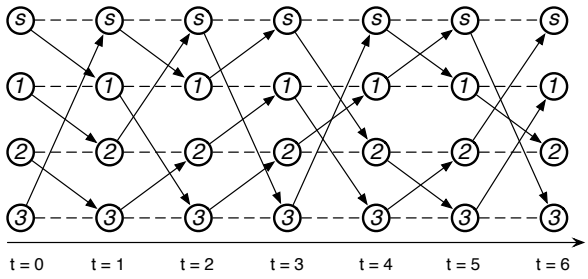

Fig. 2. A trellis graph of 4 nodes. Solid edges model the actual block transmissions, while dashed edges model information accumulation in the same node along the time line.

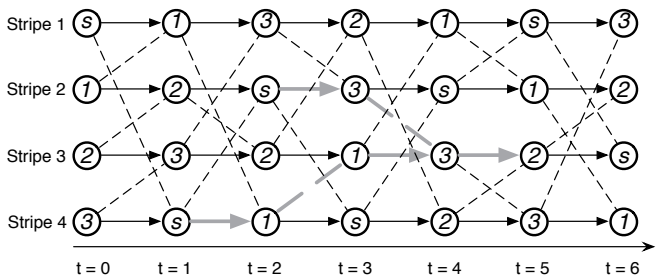

Fig. 3. An equivalent graph of the trellis in Fig. 2. Bold highlighted lines ($s, 3, 3, 2$ and $s, 1, 1, 3$) are two examples of the cross-stripe paths of the type described in (12).

[13]. Furthermore, according to [12], random network coding achieves this upper bound with high probability for a sufficiently large field size.

If in each slot, the sender-receiver pairs form a random matching between all the upload and download ports of the nodes such that no node is uploading to itself, we can thus transform the trellis in Fig. 2 into an equivalent trellis in Fig. 3, which consists of $N+1$ stripes of solid edges with cross-stripe dashed edges. Let us consider a particular node v and analyze $\text{maxflow}(v_t)$ at time t. Two paths in Fig. 3 are considered *edge-disjoint*, if they do not share any solid edge of capacity 1. Note that a path can only go from the left to the right. A max-flow from s_0 to v_t equals to the number of all edge-disjoint paths, each from a certain s_{τ_1} to a certain v_{τ_2} for $0 \le \tau_1 < \tau_2 \le t$.

For each s_τ in Fig. 3, there is a path of solid edges from s_τ to the first v after s_τ in the same stripe. We call such a v a *descended* v, *e.g.*, node 1 at time 1, such an s a *descending* s, and such a path a *descended path* associated with node v. If a v is not descended, we call it a *free* v, *e.g.*, node 1 at time 4. And if an s is not descending, we call it a *free* s. Clearly, all the descended paths for node v are edge-disjoint from each other. The question is, how many more edge-disjoint paths into free v's can we find?

For a free v at time $\tau_2 \le t$, $v_{\tau_2}^{free}$, we let it trace back to some $s_{\tau_1}^{free}$ in a different stripe via a third node u, and consider the cross-stripe path:

$$p = s_{\tau_1}^{free} \to u_{\tau_1+1} \to u_{\tau_1+2} \to \ldots \to u_{\tau_2-1} \to v_{\tau_2}^{free}, \tag{12}$$

such that $\tau_1 + 1 < \tau_2 - 1$. Note that u_{τ_1+1} must succeed $s_{\tau_1}^{free}$ in the same stripe, while u_{τ_2-1} must precede $v_{\tau_2}^{free}$ in the same stripe. u_{τ_1+1} can be connected to u_{τ_2-1} via u's through dashed edges. The examples of such cross-stripe paths are the grey paths $s_2 \to 3 \to 3 \to 2$ and $s_1 \to 1 \to 1 \to 3$ in Fig. 3.

Now we conjecture that for every $u_{\tau_2-1}, v_{\tau_2}^{free}$ pair in one stripe, there is a corresponding pair $s_{\tau_1}^{free}, u_{\tau_1+1}$ in another stripe with $\tau_1 + 1 < \tau_2 - 1$, and vice versa. If this is true, for every $v_{\tau_2}^{free}$, we can find a cross-stripe path $s_{\tau_1}^{free} \to u_{\tau_1+1} \to \ldots \to u_{\tau_2-1} \to v_{\tau_2}^{free}$ that is edge-disjoint from all other cross-stripe or descended paths into v's.

For example, if v is node 2, then s_2 is a free s w.r.t. node 2, and can thus initiate a cross-stripe path into the free node 2 at time 5: $s_2 \to 3 \to 3 \to 2$. This path is edge-disjoint from any descended paths into node 2 in stripe 2 or stripe 3, and from any other cross-stripe paths into free node 2's of the same kind. We will show that by the law of large numbers, for large t this conjecture holds with high probability. At time t, there are t node v's in total in the trellis of Fig. 3. Since each of them is the tail of an edge-disjoint path headed by a different s_τ, $\text{maxflow}(v_t)$ is almost t, leading to a download rate of $r_v = \lim_{t\to\infty} \text{maxflow}(v_t)/t = 1$.

B. Proving Theorem 1

The following lemma plays a key role in proving Theorem 1.

Lemma 2: Corollary of Strong Law of Large Numbers: Let $X_1^{(1)}, X_2^{(1)}, \ldots$ and $X_1^{(2)}, X_2^{(2)}, \ldots$ be two sequences of IID random variables following the same distribution with a finite mean $\overline{X}$. The two sequences are not necessarily independent from each other. Let $S_n^{(1)} = \sum_{i=1}^n X_i^{(1)}$ and $S_n^{(2)} = \sum_{i=1}^n X_i^{(2)}$. Then for any $\epsilon > 0$, $\delta > 0$, there is an integer $n_0(\epsilon, \delta)$ such that

$$\Pr\Big(\bigcap_{m \ge n_0} (S_m^{(1)} - S_m^{(2)} \le m\epsilon)\Big) \ge 1 - \delta. \tag{13}$$

Proof: By the strong law of large numbers [14], for any $\epsilon' > 0$ and $\delta' > 0$, there is an integer $n_0(\epsilon', \delta')$ such that

$$\Pr\Big(\sup_{m \ge n_0} \Big|\frac{S_m^{(i)}}{m} - \overline{X}\Big|\Big) > \epsilon') \le \delta', \quad i = 1, 2. \tag{14}$$

By the union bound,

$$\Pr\Big(\sup_{m \ge n_0} \Big|\frac{S_m^{(1)}}{m} - \overline{X}\Big| > \epsilon' \bigcup \sup_{m \ge n_0} \Big|\frac{S_m^{(2)}}{m} - \overline{X}\Big| > \epsilon'\Big) \le 2\delta'.$$

Then we have

$$\begin{aligned}
&\Pr\Big(\bigcap_{m \ge n_0} (S_m^{(1)} - S_m^{(2)} \le 2m\epsilon')\Big) \\
&\ge \Pr\Big(\bigcap_{m \ge n_0} (|S_m^{(1)} - S_m^{(2)}| \le 2m\epsilon')\Big) \\
&\ge \Pr\Big(\sup_{m \ge n_0} \Big|\frac{S_m^{(1)}}{m} - \overline{X}\Big|\Big) \le \epsilon' \bigcap \sup_{m \ge n_0} \Big|\frac{S_m^{(2)}}{m} - \overline{X}\Big|\Big) \le \epsilon') \\
&\ge 1 - 2\delta'
\end{aligned} \tag{15}$$

Setting $\delta = 2\delta'$ and $\epsilon = 2\epsilon'$, we have proved the lemma. □

Proof of Theorem 1: We first prove (7) for the case of random sender-receiver matching. Let us consider a particular node v and characterize maxflow(v_t) at time t. In each slot, the sender-receiver pairs form a random matching between all the upload and download ports of the nodes, such that no node is uploading to itself. As a result, each stripe in Fig. 3 forms a random walk with the same transition matrix $\boldsymbol{P} = [p_{ij}]$:

$$p_{ij} = \begin{cases} 1/N, & i,j \in V, i \neq j, \\ 0, & i,j \in V, i = j. \end{cases} \tag{16}$$

Apparently, stripes $2, \ldots, N+1$ are delayed renewal processes of stripe 1 headed by s. Let us first consider the v's in stripe 2. Since each descended v in stripe 2 is the tail of an edge-disjoint descended path, we want to know whether each free v in stripe 2 is also the tail of an edge-disjoint path.

Consider the renewal process of visiting a free s followed by $u \in V \backslash \{s, v\}$ in stripe 1, $\{N^{(1)}(t); t \geq 0\}$, with inter-renewal intervals $\{X_n^{(1)}; n \geq 1\}$ and renewal epochs $\{S_n^{(1)}; n \geq 1\}$. Consider another renewal process of visiting a free v preceded by the same intermediate node u in stripe 2, $\{N^{(2)}(t); t \geq 0\}$, with inter-renewal intervals $\{X_n^{(2)}; n \geq 1\}$ and renewal epochs $\{S_n^{(2)}; n \geq 1\}$. A v in stripe 2 is a free v with probability

$$1 - (\frac{1}{N} + \frac{N-1}{N} \cdot \frac{1}{2}) = \frac{N-1}{2N}$$

which is the probability given this v, another v appears in stripe 2 before any s. Similarly, an s in stripe 1 is a free s with the same probability $(N-1)/2N$. Furthermore, a free s is followed by a node $u \in V \backslash \{s, v\}$ in stripe 1 with probability $1/(N-1)$, which equals to the probability that a free v is preceded by u in stripe 2.

Since the process of visiting v in stripe 2 is a delayed renewal process of visiting s in stripe 1, $\{N^{(2)}(t); t \geq 0\}$ is also a delayed renewal process of $\{N^{(1)}(t); t \geq 0\}$. Thus, $\{X_n^{(1)}; n \geq 1\}$ and $\{X_n^{(2)}; n \geq 2\}$ are two identical (not independent) IID sequences, with

$$\mathbf{E}[X_n^{(1)}] = \mathbf{E}[X_n^{(2)}] = (N+1) \cdot \frac{2N}{N-1} \cdot (N-1). \tag{17}$$

By Lemma 2, for any arbitrarily small $\epsilon > 0$ and $\delta > 0$, and a sufficiently large integer k_0, there is an integer $n_0(\epsilon, \delta)$ such that

$$\Pr(\bigcap_{n_0 \leq m \leq k_0} (S_m^{(1)} - S_m^{(2)} \leq m\epsilon)) \geq 1 - \delta. \tag{18}$$

If we delay the starting time of $\{N^{(2)}(t); t \geq 0\}$ by $\Delta t = k_0\epsilon + 3$ and obtain another renewal process $\{N'^{(2)}(t); t \geq \Delta t\}$, where

$$N'^{(2)}(t) = N^{(2)}(t) - N^{(2)}(k_0\epsilon + 3). \tag{19}$$

Then for each integer $m \in [n_0, k_0]$, the m^{th} renewal epoch in $\{N'^{(2)}(t); t \geq k_0\epsilon + 3\}$ is greater than the m^{th} renewal epoch in $\{N^{(1)}(t); t \geq 0\}$ by at least 3 time slots w.h.p.

As a result, by time $S_{k_0}^{(2)} + k_0\epsilon + 3$, w.h.p. at least $k_0 - n_0$ pairs of $s_{\tau_1}^{free}, u_{\tau_1+1}$ in stripe 1 have a one-to-one mapping with $u_{\tau_2-1}, v_{\tau_2}^{free}$ pairs in stripe 2 so that $\tau_1 + 1 < \tau_2 - 1$. And this is true for any intermediate node $u \in V \backslash \{s, v\}$. Hence, we have shown that the contribution to the maxflow(v_t) by free v's in stripe 2 flow$_2(v_t^{free})$ satisfies

$$\begin{aligned} \lim_{t\to\infty} \frac{\text{flow}_2(v_t^{free})}{t} &= \sum_{u \in V \backslash \{s,v\}} \lim_{k_0 \to \infty} \frac{k_0 - n_0}{S_{k_0}^{(2)} + k_0\epsilon + 3} \\ &\to (N-1) \cdot \frac{1}{\mathbf{E}[X_n^{(2)}]} \quad \text{with probability 1} \\ &= \frac{1}{N+1} \cdot \frac{N-1}{2N} \end{aligned} \tag{20}$$

As each descended v in stripe 2 incurs an edge-disjoint descended path, we have the contribution from descended v's

$$\lim_{t\to\infty} \frac{\text{flow}_2(v_t^{des})}{t} = \frac{1}{N+1} \cdot (1 - \frac{N-1}{2N}). \tag{21}$$

Since the renewal process of visiting a node $v \in V$ in each stripe is a delayed renewal process of visiting v in stripe 1, we can pair up the free v's in stripe i with free s's in stripe $i-1$ in a similar way, for $i = 2, \ldots, N+1$ (stripe 1 can be paired up with stripe $N+1$). Therefore,

$$\begin{aligned} r_v &= \lim_{t\to\infty} \frac{\text{maxflow}(v_t)}{t} \\ &= \sum_{i=1}^{N+1} (\lim_{t\to\infty} \frac{\text{flow}_i(v_t^{free})}{t} + \lim_{t\to\infty} \frac{\text{flow}_i(v_t^{des})}{t}) = 1. \end{aligned} \tag{22}$$

To prove (8) for the case of random receiver selection, we notice that if each node $v \in V$ randomly selects another node $u \in V$ that has not been chosen in this slot, only the last node may have to upload to itself. The probability that v is the last node to choose its receiver in this slot is $1/(N+1)$. Given that v chooses its receiver last, the probability that v has to choose itself is the probability that none of the previous N nodes chooses v as its receiver, which is at most

$$(1 - \frac{1}{N}) \prod_{j=2}^{N} \left(1 - \frac{1}{N - (j-2)}\right) = \frac{N-1}{N^2} < \frac{1}{N}, \tag{23}$$

since the first node fails to choose v with probability $1 - \frac{1}{N}$, and conditioned on that all previous nodes did not choose v, the j^{th} node fails to choose v with probability at most $1 - \frac{1}{N-(j-2)}$, $2 \leq j \leq N$. Hence, the probability that a node v is preceded by another v in the same stripe is at most $\frac{1}{N(N+1)}$, which is also the probability that an s immediately precedes another s. These v's and s's do not contribute to the maximum flow, *i.e.*, at most $\frac{1}{N(N+1)}$ of all v's or s's cannot be the tails or heads of any edge-disjoint paths. Thus, there is at most a $\frac{1}{N(N+1)}$ loss on the download rate of each node, proving (8). □

C. Heterogeneous Node Capacity

We now prove Theorem 2. For a heterogeneous network with nodes V, each $v \in V$ with upload capacity U_v can be decomposed into U_v child nodes $\mathcal{C}(v)$, each of unit upload and download capacity. The entire heterogeneous network is thus transformed into a homogeneous network of nodes $V' = \bigcup_{v \in V} \mathcal{C}(v)$, where $|V'| = \sum_{v \in V} U_v$ and $U_{v'} = D_{v'} = 1$, $\forall v' \in V'$. Let $P(v') \in V$ denote the parent node of $v' \in V'$.

Random rate allocation now translates to the following: in each time slot, following a random order, each node $u' \in V'$ randomly selects another node $v' \in V'$ that has not been chosen in this slot as its receiver. According to the preceding analysis, if each $u' \in V'$ always chooses a $v' \in V'$ such that $P(u') \neq P(v')$, then each $v' \in V'$ has an asymptotic download rate of $r_{v'} = 1$ in the transformed network. However, each $u' \in V'$ chooses a $v' \in V'$ that has the same parent with probability

$$|\mathcal{C}(P(u'))|/|V'| = U_{P(u')}/\sum_{v\in V} U_v. \tag{24}$$

This contributes to the fraction of loss in the download rate of $P(u')$. Similar to the proof of (8), we have the result that the asymptotic download rate of each parent node $v \in V\backslash\{s\}$ is

$$\begin{aligned} r_v &= \sum_{v'\in\mathcal{C}(v)} r_{v'} = \sum_{v'\in\mathcal{C}(v)} 1\cdot\left(1-\frac{U_{P(v')}}{\sum_{v\in V}U_v}\right) \\ &= U_v\left(1-\frac{U_v}{\sum_{v\in V}U_v}\right), \end{aligned} \tag{25}$$

proving Theorem 2.

VI. The Performance Lower Bound

We now let both N and k scale and prove the performance lower bound given in Theorem 3. We first prove an intermediate result outlined in the following theorem, the proof of which makes use of Lemma 3 and Lemma 4.

Theorem 4: Assume $U_v = 1 \leq D_v < \infty$ for all $v \in V$. Assume the source blocks form a stream, *i.e.*, $k \to \infty$. If random receiver selection is applied with random network coding, as N scales, for any constants $c > 0$ and $0 < \delta < 1$, by time

$$t = cN(N+1)(1+2\delta) + (N+1)\ln(N+1), \tag{26}$$

the following bound holds for all $v \in V\backslash\{s\}$:

$$\mathbf{E}[\text{maxflow}(v_t)] > \frac{1}{2}t + (\frac{1}{2} - \mathrm{e}^{-1})cN^2 + o(N^2). \tag{27}$$

Lemma 3: Chernoff Bounds for Exponential Random Variables: Let $X_1, X_2, \ldots, X_n$ be IID exponential variables such that $\Pr(X_i \leq x) = 1 - \mathrm{e}^{-\lambda x}$, $x \geq 0$. Let $S_n = \sum_{i=1}^n X_i$ and $\mu_n = \mathbf{E}[S_n] = n/\lambda$. Then for any $\delta > 0$,

$$\Pr(S_n > (1+\delta)\mu_n) < \left(\frac{\mathrm{e}^{\delta}}{1+\delta}\right)^{-n}. \tag{28}$$

For any $0 < \delta < 1$,

$$\Pr(S_n < (1-\delta)\mu_n) < \left(\frac{\mathrm{e}^{-\delta}}{1-\delta}\right)^{-n}. \tag{29}$$

Proof: The proof follows the standard practice of deriving Chernoff bounds [15] and is omitted here. □

Lemma 4: Let $X_1^{(1)}, X_2^{(1)}, \ldots$ and $X_1^{(2)}, X_2^{(2)}, \ldots$ be two sequences of IID random variables following the same exponential distribution with mean $1/\lambda$. The two sequences are not necessarily independent from each other. Let $S_n^{(1)} = \sum_{i=1}^n X_i^{(1)}$ and $S_n^{(2)} = \sum_{i=1}^n X_i^{(2)}$. Let $\mu_n = \mathbf{E}[S_n^{(1)}] = \mathbf{E}[S_n^{(2)}] = n/\lambda$. Then for any $0 < \delta < 1$ and any integer $n \geq 1$,

$$\Pr(S_n^{(1)} - S_n^{(2)} \geq 2\delta\mu_n) < \left(\frac{\mathrm{e}^{\delta}}{1+\delta}\right)^{-n} + \left(\frac{\mathrm{e}^{-\delta}}{1-\delta}\right)^{-n} \tag{30}$$

Proof: If $S_n^{(1)} \leq (1+\delta)\mu_n$ and $S_n^{(2)} \geq (1-\delta)\mu_n$, then $S_n^{(1)} - S_n^{(2)} \leq 2\delta\mu_n$. Hence, we have

$$\begin{aligned} &\Pr(S_n^{(1)} - S_n^{(2)} \leq 2\delta\mu_n) \\ &\geq \Pr(S_n^{(1)} \leq (1+\delta)\mu_n \bigcap S_n^{(2)} \geq (1-\delta)\mu_n) \\ &\geq 1 - \left(\frac{\mathrm{e}^{\delta}}{1+\delta}\right)^{-n} - \left(\frac{\mathrm{e}^{-\delta}}{1-\delta}\right)^{-n}, \end{aligned} \tag{31}$$

where the second inequality is due to the union bound and Lemma 3. □

Proof of Theorem 4: Considering a particular receiver node $v \in V\backslash\{s\}$, we have

$$\mathbf{E}[\text{maxflow}(v_t)] = \mathbf{E}[\text{flow}(v_t^{des})] + \mathbf{E}[\text{flow}(v_t^{free})]. \tag{32}$$

Referring to Fig. 3, by time t, the number of v's that appear in the trellis is t. Each v is a descended v with probability $(N+1)/2N \to \frac{1}{2}$. Thus, for large N,

$$\mathbf{E}[\text{maxflow}(v_t)] = \frac{t}{2} + \mathbf{E}[\text{flow}(v_t^{free})]. \tag{33}$$

Now we bound the flow contributed by free v's $\mathbf{E}[\text{flow}(v_t^{free})]$. By time

$$t_1 = (N+1)\ln(N+1), \tag{34}$$

the probability that s does not appear in a certain stripe is less than

$$(1-\frac{1}{N+1})^{(N+1)\ln(N+1)} \leq e^{-\ln(N+1)} = \frac{1}{N+1}$$

Thus, at time t_1, the expected number of stripes that contain at least one s is at least N. In other words, starting from time $t_1 = (N+1)\ln(N+1)$, the free s's in these N stripes can contribute to the flow received by free v's. Since the behavior of the trellis in $(t_1, t]$ is independent of its behavior in $[0, t_1)$, to compute $\mathbf{E}[\text{flow}(v_t^{free})]$, we can consider the flow contribution of each stripe, multiplied by the expected number of stripes, N, that can contribute to the flow received by free v's starting from time t_1.

We aim to show that at time $t = cN(N+1)(1+2\delta) + (N+1)\ln(N+1)$,

$$\mathbf{E}[\text{flow}(v_t^{free})] > (\frac{1}{2} - \mathrm{e}^{-1})cN^2 + o(N^2). \tag{35}$$

Use S_j to denote the set of all the j^{th} free s's in each stripe starting from time $t_1 = (N+1)\ln(N+1)$. Use V_j to denote the set of all the j^{th} free v's in each stripe starting from time

$$t_2 = (N+1)\ln(N+1) + 2\delta cN(N+1) + 3. \tag{36}$$

Let $j_{\max}$ denote the maximum j such that the timestamps of every $s \in S_j$ and every $v \in V_j$ are no more than t. Recall that an element $s \in S_j$ can be matched with an element $v \in V_j$ to form an edge-disjoint path, if

1. s is succeeded by a node u in its own stripe while v is preceded by the same node u in its own stripe;
2. the timestamp of s is smaller than the timestamp of v by at least 3 time slots.

Let $\mathcal{M}_j$ be a matching between the elements in S_j and the elements in V_j so that conditions 1 and 2 are satisfied. Clearly, we have

$$\mathbf{E}[\text{flow}(v_t^{free})] \geq \mathbf{E}\Big[\sum_{j=1}^{j_{\max}} |\mathcal{M}_j|\Big]. \quad (37)$$

The roadmap of the rest of the analysis is described as follows. First, we will show that the number of free s's in each stripe during the interval $[t_1, t-2\delta cN(N+1)]$ equals to the number of free v's in each stripe during the interval $[t_2, t]$, and equals to $\frac{1}{2}cN + o(N)$. If this is true, and if conditions 1 and 2 were ignored, considering N stripes, the total number of (s,v) pairs in $\bigcup_{1\leq j\leq j_{\max}} \mathcal{M}_j$ would be at least $\frac{1}{2}cN^2 + o(N^2)$. Furthermore, let us define L_1 and L_2 as the numbers of (s,v) pairs in $\bigcup_{1\leq j\leq j_{\max}} \mathcal{M}_j$ that violate conditions 1 and 2, respectively. By the union bound, we can then obtain

$$\mathbf{E}\Big[\sum_{j=1}^{j_{\max}} |\mathcal{M}_j|\Big] \geq \frac{1}{2}cN^2 + o(N^2) - \mathbf{E}[L_1] - \mathbf{E}[L_2]. \quad (38)$$

We will give upper bounds for $\mathbf{E}[L_1]$ and $\mathbf{E}[L_2]$, so that we can obtain (35) by combining (37) and (38).

Let $\{N^s(t); t \in [t_1, t-2\delta cN(N+1)]\}$ denote the process in which free s's appear in a certain stripe during time interval $[t_1, t-2\delta cN(N+1)]$. For any $l > 0$, the inter-arrival time Y between s's satisfies

$$\Pr(Y \leq l(N+1)) = 1-(1-\frac{1}{N+1})^{l(N+1)} \to 1-\mathrm{e}^{-l}. \quad (39)$$

This means $\frac{Y}{N+1}$ is asymptotically exponential with mean 1. Since each s is a free s with probability $\frac{N-1}{2N} \to 1/2$, the inter-arrival time X between free s's in a stripe satisfies that $X' = \frac{X}{N+1}$ is asymptotically exponential with mean 2 [14]. Let $\tau = \frac{t}{N+1}$ and define a rescaled process of $N^s(t)$ as

$$N'^s(\tau) = N^s(\tau(N+1)),\ \tau \in [\frac{t_1}{N+1}, \frac{t}{N+1} - 2\delta cN]. \quad (40)$$

Then $N'^s(\tau)$ is a Poisson process with mean inter-arrival time $\overline{X}' = 2$.

According to the strong law for renewal processes (pp. 60 [14]), with probability 1, the number of free s's in process $N'^s(\tau)$ during the interval $[\frac{t_1}{N+1}, \frac{t}{N+1} - 2\delta cN]$ is

$$\frac{\frac{t}{N+1} - 2\delta cN - \frac{t_1}{N+1}}{\overline{X}'} + o(\frac{t}{N+1} - 2\delta cN - \frac{t_1}{N+1})$$
$$= \frac{1}{2}cN + o(N). \quad (41)$$

Similarly, the number of free v's in each stripe during the interval $[t_2, t]$ is also $\frac{1}{2}cN + o(N)$ with probability 1. As N stripes are considered, the expected number of (s,v) pairs in $\mathcal{M}_j$ given that condition 1 and 2 are ignored is at least

$$\frac{1}{2}cN^2 + o(N^2). \quad (42)$$

Next, we consider condition 1 and bound $\mathbf{E}[L_1]$ from above. Since each $s \in S_j$ must have a different timestamp in the trellis, each $s \in S_j$ is independently succeeded by $u \in V\backslash\{s\}$ with probability $1/N$. View the succeeding nodes u's as bins, and $s \in S_j$ as balls, we have a classical problem of throwing $|S_j|$ balls into N bins [15]. The expected number of empty bins is

$$(1-\frac{1}{N})^N N \to \mathrm{e}^{-1}N \quad \text{as } N \to \infty. \quad (43)$$

In other words, on expectation, $\mathrm{e}^{-1}N$ distinct nodes in $V\backslash\{s\}$ do not succeed any $s \in S_j$. Similarly, on expectation, $\mathrm{e}^{-1}N$ distinct nodes in $V\backslash\{v\}$ do not precede any v in V_j. In the worst case, a matching $\mathcal{M}_j$ of size at least $(1-2\mathrm{e}^{-1})N$ can be formed between $s \in S_j$ and $v \in V_j$ so that condition 1 is satisfied. Hence,

$$\mathbf{E}[L_1] < 2\mathrm{e}^{-1}N \cdot (\frac{1}{2}cN + o(N)). \quad (44)$$

Finally, we consider condition 2 and bound $\mathbf{E}[L_2]$ from above. Recall that the time-rescaled process of visiting free s's in each stripe $N'^s(\tau)$ is a Poisson process. Since $t_2 - t_1 = 2\delta c(N+1) + 3$, by Lemma 4, each (s,v) pair in $\mathcal{M}_j$ violates condition 2 with probability at most

$$\Big(\frac{\mathrm{e}^{\delta}}{1+\delta}\Big)^{-j} + \Big(\frac{\mathrm{e}^{-\delta}}{1-\delta}\Big)^{-j}. \quad (45)$$

Let $\beta_1 = \mathrm{e}^{\delta}/(1+\delta)$ and $\beta_2 = \mathrm{e}^{-\delta}/(1-\delta)$. Considering N stripes, we thus have

$$\mathbf{E}[L_2] < N\sum_{j=1}^{j_{\max}} (\beta_1^{-j} + \beta_2^{-j}) < N\Big(\frac{\beta_1^{-1}}{1-\beta_1^{-1}} + \frac{\beta_2^{-1}}{1-\beta_2^{-1}}\Big)$$
$$= O(N). \quad (46)$$

Combining (37), (38), (42), (44), (46), we obtain (35). Combining (33) and (35), we have arrived at (27) and proved the theorem. □

We can now prove Theorem 3 based on Theorem 4. As long as k is large enough and network coding is applied, by Lemma 1, $B_v(t) = \text{maxflow}(v_t)$. By Theorem 4, if t obeys the scaling in (26), we have that

$$\lim_{N\to\infty} \frac{\mathbf{E}[B_v(t)]}{t} > \frac{1}{2} + \frac{\frac{1}{2} - \mathrm{e}^{-1}}{1+2\delta} \quad (47)$$

holds for any arbitrarily small constant $\delta > 0$. This is equivalent to $\lim_{N\to\infty} \frac{\mathbf{E}[B_v(t)]}{t} \geq 1 - \mathrm{e}^{-1}$, proving the bound in Theorem 3.

VII. Simulation Results

We conduct simulations to substantiate and complement our theoretical observations. We first consider a homogeneous network of size $|V| = 100$ and broadcast k blocks from a single source to $N = 99$ receiver nodes, using random receiver selection with random network coding done in $GF(2^8)$. Each node has $U_v = D_v = 1$. In Fig. 4, we vary k and for each k, plot $\sum_{i=1}^{N} B_i(t)/N$ as an estimate of $\mathbf{E}[B_v(t)]$ in the network versus time t, until the point at which all the nodes finish downloading (represented by a big dot for each k). We also

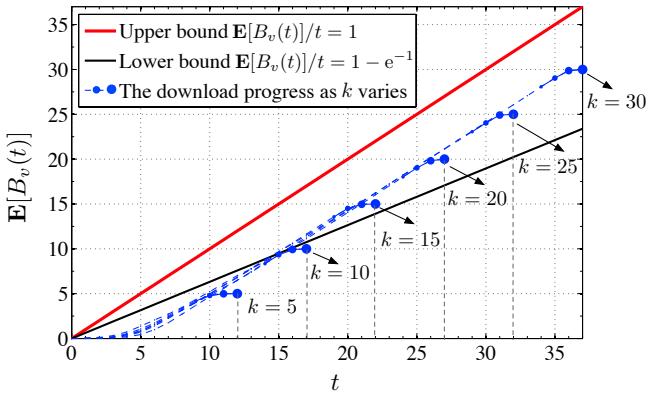

Fig. 4. The average download progress when $N = 99$ as k varies. For each k, $\mathbf{E}[B_v(t)]$ is plotted until the point that all the nodes have finished downloading (represented by the biggest dot on the line of that k).

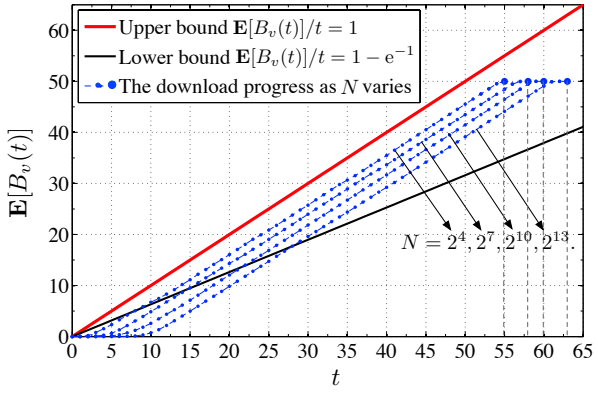

Fig. 5. The average download progress when $k = 50$ as N varies. For each N, $\mathbf{E}[B_v(t)]$ is plotted until the point that all the nodes have finished downloading (represented by the biggest dot on the line of that N).

plot the obvious upper bound $\mathbf{E}[B_v(t)]/t = 1$ and the lower bound $\mathbf{E}[B_v(t)]/t = 1 - \mathrm{e}^{-1}$ inspired by Theorem 3.

We derive several useful observations from Fig. 4. First, we find as k increases to a large value with N fixed, $\mathbf{E}[B_v(t)]/t$ approaches 1 for sufficiently large t before all the nodes finish downloading. Second, recall that $\max_v T_v(k)$ is the finish time of the last node. We observe that $\max_v T_v(k) = k + 7 = k + \lceil \log_2 N \rceil$ for all k's in Fig. 4, which is the lowest possible broadcast finish time by (5). The same observation holds for larger k's. Hence, we have $\lim_{k\to\infty} k/\max_v T_v(k) \to 1$, substantiating the correctness of Theorem 1. It is also interesting to note that the slow download rates at the beginning and near the end account for the delay $\lceil \log_2 N \rceil$ in broadcast finish time, while in the middle the download rate of each node is almost 1. Third, we observe that with a sufficiently large t, $\mathbf{E}[B_v(t)]/t$ will always fall above the lower bound outlined by Theorem 3 even if $k = \Omega(N^2)$ does not hold. Therefore, for any ranges of N and k values, we can be much more optimistic about the practical performance of random receiver selection with network coding than the lower bound proved in Theorem 3 with theoretical rigor.

We further fix k to 50, and plot $\mathbf{E}[B_v(t)]$ over time t during the download process in Fig. 5 for different N's. We observe that as N varies from 16 to 8192, the broadcast finish time increases from 55 to only 63. This confirms that with network coding allowed, a protocol as simple as random receiver selection behaves near the optimal protocol, achieving an asymptotic download rate of 1 per node and a broadcast finish time close to the optimal value $k + \lceil \log_2 N \rceil$.

VIII. Concluding Remarks

Motivated by content dissemination in P2P networks, we consider the problem of broadcasting k blocks to N nodes from a single source, with both node upload and download capacity constraints. We prove that in homogeneous networks with coding allowed, simple randomized receiver selection can achieve optimal download rates as k scales. In heterogeneous networks, optimal and fair download rates at the nodes can be approximately achieved by randomized rate allocation with network coding for a large k. We also give a performance lower bound of coded randomized broadcast when both k and N scale. Results from this paper reveal that network coding not only eases block scheduling in P2P content dissemination, but also enables the use of simple randomized broadcast protocols to achieve optimal performance, which was only previously known to be achievable with elaborate tree construction or protocols with non-trivial node state exchanges.

References

[1] D. M. Chiu, R. W. Yeung, J. Huang, and B. Fan, "Can Network Coding Help in P2P Networks?" in *Proc. International Workshop on Network Coding*, Boston, April 2006.

[2] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An Efficient Mechanism for Content Distribution in a Peer-to-Peer (P2P) Network," in *Proc. of ACM SIGCOMM Asia Workshop*, Beijing, China, April 2005.

[3] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, 2007.

[4] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized Decentralized Broadcasting Algorithms," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, May 2007.

[5] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility Maximization in Peer-to-Peer Systems," in *Proc. ACM SIGMETRICS*, Annapolis, Maryland, USA, June 2008.

[6] S. Deb, M. Médard, and C. Choute, "Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2486–2507, June 2006.

[7] R. W. Yeung, "Avalanche: A Network Coding Analysis," *Communications in Information and Systems*, vol. 7, no. 4, pp. 353–358, 2007.

[8] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with Multiple Messages," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, 2007.

[9] J. Mundinger, R. Weber, and G. Weiss, "Optimal Scheduling of Peer-to-Peer File Dissemination," *Journal of Scheduling*, 2007.

[10] A. Bar-Noy and S. Kipnis, "Broadcasting Multiple Messages in Simultaneous Send/Receive Systems," *Discrete Applied Mathematics*, vol. 55, pp. 95–105, 1994.

[11] C. Feng, B. Li, and B. Li, "Understanding the Performance Gap between Pull-based Mesh Streaming Protocols and Fundamental Limits," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, April 19-25 2009.

[12] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," in *Proc. IEEE Int'l Symp. Information Theory (ISIT)*, 2003.

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

[14] R. Gallager, *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1996.

[15] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.