Fair and distributed peer-to-peer allocation of a common, refillable resource

Sachin Agarwal, Moshe Laifenfeld, Andrew Hagedorn, Ari Trachtenberg and Murat Alanyali

Abstract

We consider the general problem of distributed and fair peer-to-peer (p2p) allocation of a common, refillable resource. This problem recurrs in a number of scenarios, for example grid computing, content distribution, Internet Service Provider service sharing, and file sharing over asymmetric channels. We present several distributed schemes for this allocation problem, and show that these schemes guarantee two key properties: (i) asymptotic fairness, in that (even maliciously colluding) users are proportionally assigned resources corresponding to what they contribute; (ii) natural incentive to join and cooperate fairly in the system. We demonstrate the practicability of our approaches on a prototype p2p file sharing system designed for typical residential internet connections, in which download capacities often significantly exceed upload capacities. Our implementation shares file data when communications are idle using random linear codes, so that, when needed, an end-user can download a file from several sources at a higher data rate than his home computer's upload capacity. We present experimental results that support our analytical guarantees.

A version of this article appeared as:

- S. Agarwal, M. Laifenfeld, A. Hagedorn, A. Trachtenberg and M. Alanyali, “Fair and distributed peer-to-peer allocation of a common, refillable resource”, ACM Journal on Parallel and Distributed Computation, 69:12, pp. 974-988

I. Introduction

In this paper we focus on a variation of the distributed resource allocation problem, which has a variety of applications, including bandwidth sharing and content distribution in peer-to-peer networks, service sharing among local Internet Service Providers (ISPs) and grid computing, to name a few (36; 27). Our problem is formulated as follows: a heterogeneous collection of peers demand the occasional use of a given resource, which each of them individually have through a limited, but refillable, allotment. *limited* and *refillable* resource, Each peer's *experience*, which it tries to maximize, depends on the time needed to obtain the resource upon demand. As such, when a peer is operating in isolation, its experience is limited by its resource capacity in a straightforward manner. However when peers team-up to form a network, each peer could potentially enhance its experience by utilizing unused capacity of other idle peers (see Figure 1).

A. Application 1: grid computing

To take a concrete example of our model, consider the case of grid computing. In this scenario, a large number of stand-alone computers are connected via a network, such as an Ethernet, in order to perform massive computations or process a large volume of data in a distributed manner (35). In the context of our model, each computer has a limited and refillable CPU processing capability by virtue of the fact that each CPU can perform one new task at each time step. The computers spend most of their life idling (and avilable for peers on the network), and the experience of a particular user relates to the time needed to perform his task when needed.

To date, most grid computing projects are operated in a controlled environment over privately-owned networks (*e.g.*, Google's data-centers, NSA, NASA). Alternatively, *volunteer* computing grids are typically dedicated to specific research problems (*e.g.*, distributed.net, SETI@home), and they are thus controlled and programmed by only a few peers. However, one may readily envision grid computing in a peer-to-peer (p2p) environment, allowing peers to share their computer resources over a public network such as the Internet for individual benefit. Such networks, however, would have to attract participating peers (with typically idle computer resources) by providing an incentive to join the network. In addition they would have to provide guarantees even in the presence of malicious peers trying to disrupt or manipulate the network's performance.

B. Application 2: File sharing

Another application of our allocation model concerns p2p file sharing over asymmetric channels.

Many users connect to the Internet through asymmetric links in which the upload transfer capacities are significantly smaller than download capacities. Internet Service Providers (ISPs) employ this asymmetric design based on the premise that casual Internet use mostly involves downloading content from a relatively small number of *content providers*, such as large data portals, mail servers, web servers, or the like. For example, carrier-less amplitude/phase (CAP) DSL allocates transmission frequencies between 25KHz and 160KHz for uploads, and frequencies from 240KHz to 1500KHz for downloads, making downloads significantly faster than uploads. Asymmetric channels abound in various other practical scenarios, including analog modem dial-up, wireless cellular Internet connections, and heterogeneous sensor networks.

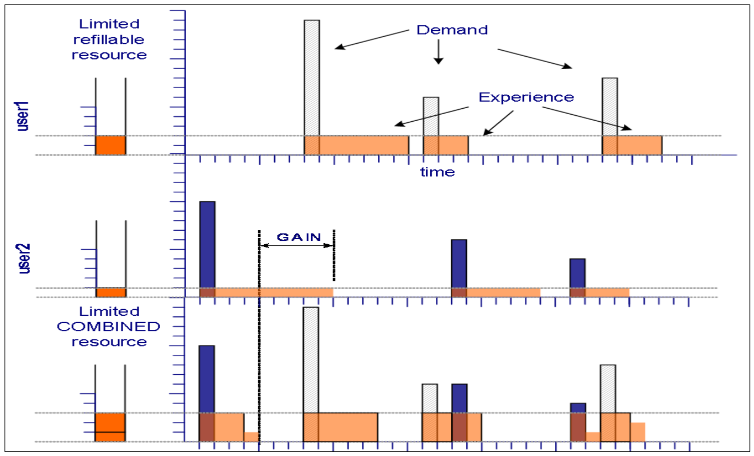


Fig. 1. Examples of the general resource allocation model.

Recently the ‘mostly download’ profile of users has started to change. Users now commonly have access to devices like digital video cameras, high resolution scanners, high capacity sound recorders, and other audio/video devices that capture large volumes of digital data. The fast pace of advancement in persistent storage (hard-disks, flash memory) further enables users to generate and store large volumes of diverse data on their home computers or personal web sites. This is a far call from PCs even ten years ago for, when the biggest storage requirements for home users were typically from application programs or the operating system - data that did not have to be transferred when users worked remotely.

This change in users’ access profile causes upload speed to become a bottleneck for typical remote access. Specifically, if a user wishes to remotely access data stored on a home computer, such as a song, image, or video, his access rate is limited to the minimum of the home computer’s upload capacity and the user’s current download capacity at the remote location. The asymmetry in upload and download capacity thus results in extremely poor download channel utilization at the remote host.

Resource allocation can be utilized to correct for such channel assymetries, bypassing the ‘bandwidth: use it or lose it’ service model offered by commercial ISPs. As such, users could pre-distribute important files throughout their network, and then share bandwidth (the common, refillable resource) in order to improve individual file access on demand. The network would have to maintain and reimburse ‘credit’ for contributions to the network in a fair and robust manner.

C. Goals

Our goals in this work involve developing distributed resource allocation schemes and demonstrating, by analysis and simulations, that they have the following properties:

Fairness: Idle resources are redistributed in proportion to the resources contributed by each peer to the network.

Incentive: There is a natural incentive for peers to participate and cooperate with others in the network.

Distributed operation: Only local information is needed in the implementation of the allocation scheme (i.e. no control information is exchanged and there is no central authorityx).

D. Outline

Throughout this work, we shall use example 2 above, concerning p2p file sharing, as a concrete, novel application in which to demonstrate our algorithms. In Section II we mention some of the related work in the fields of game theory, p2p systems, grid computing, and coding theory, and contrast these with our approach in Section II-A. Thereafter, we formally introduce the details of our proposed bandwidth sharing approach p2p networks in Section III. In Section IV we analytically prove the fairness of our system, which are more generally applicable to our broader problem, and show that it possesses a natural incentive for peer contributions. We simulate various aspects of our system in Section V to demonstrate its fairness and collaboration features, including specific cases where malicious peers attempt to gain unfair advantage from the system. We demonstrate the practical application of our distributed resouce allocation schemes via our implementation of the p2p system in Section VI. We also present experimental results on the encoding/decoding performance of random linear codes to demonstrate their practicability for the proposed p2p application. We conclude our findings in Section VII. Acknowledgements are provided in Section VIII.

II. RELATED WORK

Many of the studies into distributed resource allocations schemes and grid computing in particular consider market place, auction and bidding models, which require exchanging messages (bids) among peers (37; 34), and in some cases require some sort of hierarchy to establish mark place policies (34). These approaches are usually assume reasonable peer behavior, namely peers try to optimize some utility function, and therefore have no guarantees in the presence of malicious peers trying to disrupt the normal operation of the network. Our approach is geared toward a simpler barter economy scheme (38), and requires neither a hierarchy nor overhead in the form of control messages exchange among the peers.

Peer-to-peer (p2p) systems are typically used to distribute content on the Internet, and it is estimated that a major portion of the bandwidth available on consumer ISP networks carries p2p content (4). P2p services make scalable content distribution possible by utilizing peers' upload bandwidth to service other peers' download requests. It has been shown through analysis, simulations and measurements (9; 11; 8; 7; 6; 5; 10; 12; 13) that the p2p content delivery model scales gracefully with user demands for heterogeneous p2p networks. In the remainder of this section we describe some of the literature that is relevant to the different aspects of our proposed system, ending with a brief explanation of the novelty of our approach.

Content Distribution Recent works in p2p networks concentrate on mitigating non-cooperative behavior of peers by adding incentive schemes. Due to scalability issues most of these schemes are distributed, and require only local information that is readily available to each peer. Similar to our scheme, the most common schemes are based on *Barter economy*, where peers offer their bandwidth to others according to the amount of bandwidth allocated to them (2; 21; 22; 23). Although adding incentive may increase cooperation among reasonable users, namely those users that try to optimize their resources, it usually has no guarantee against malicious peers that want to disrupt others' usage of the system or gain unfair advantage at the expense of other users, and additional measures are required to reduce their effect on the network. This conclusion was recently explained in (24) where the authors explained several exploits for a selfish Bittorrent user to achieve an unfair advantage from the system.

The idea of sharing disk-space for data backup and downloads is not new. For example, the Folder-share system (1) lets users share their documents with other users. In this system, however, users only download a file from one peer, thus limiting their download speed to the upload speed of the peer. In addition the system assumes that peers do not cheat others when it comes to offering bandwidth. A different system, the Oceanstore project (3), provides a large Internet-scale data storage solution for data security and reliability using erasure codes.

Coding Random linear coding (15; 17; 12) has been used for achieving the network coding (14) min-cut bound on multicast capacity in networks. The authors in (17) proposed random linear coding as a way to avoid the coupon collector's problem (16) in a p2p storage system. Their application considers the case when *parts* of the same file are encoded and kept on separate hosts and then rebuilt. While our system can also operate in this fragmented storage mode, our emphasis is on fairness and the ability to beat the upload link bottleneck.

Analysis Much of the work on p2p systems characterizes fairness and incentives for peers to cooperate by simulations, measurements and experiments of p2p systems rather than actual analysis, probably because of the complexities arising from the size, chaotic nature and heterogeneous conditions that characterize real systems (e.g. (21; 22; 23)). Nevertheless, there has been some recent queuing theoretic analytical work (11) that uses a queuing approach to study scalability and resilience to freeloaders in p2p systems. A game theoretic approach has also been applied to the related problem of parallel downloading (*i.e.,* downloading a large file from several servers in parallel). In (25) this problem is analyzed using non-cooperative game-theoretic tools. However, this approach cannot capture the effects of malicious users.

A. Our approach

Unlike many existing p2p systems that are built for discovering and disseminating popular content, our system attempts to share *unused bandwidth* among system subscribers. Users who contribute bandwidth to the system are rewarded with higher instantaneous bandwidth availability when they need it. In all cases, users are (asymptotically) assured that all bandwidth they share with the network will be returned to them.

Our system also differs from typical p2p file-sharing systems like (2; 12) in that it is used by remote users, thus differentiating between users and network peers. More precisely, when a user u wishes to access her content (which has been distributed among the network peers off-line), she downloads content from *multiple* peers in the network, (possibly) including her own home computer. This subtle difference means that our system no longer needs the 'non-dominant' condition in (8), *i.e.*, that the upload capacity of every peer is necessarily smaller than the sum of upload capacities of all other peers. This, in turn, means that our system does not require a symmetric instantaneous 'tit-for-tat' approach to guarantee fairness (*i.e.,* the system intrinsically evens out contributions asymptotically).

Our proposed system comprises of n peers that collaborate to distribute other peers' information using their spare upload bandwidth. With each host i we associate a corresponding available upload bandwidth μ_i and available download bandwidth λ_i. Ideally, the upper bound on the download bandwidth available to peers is $\sum_{i=1\ldots n} \mu_i$, although, in practice, a user accessing the system has a limited download bandwidth that maybe smaller than the overall peer bandwidth in the system.

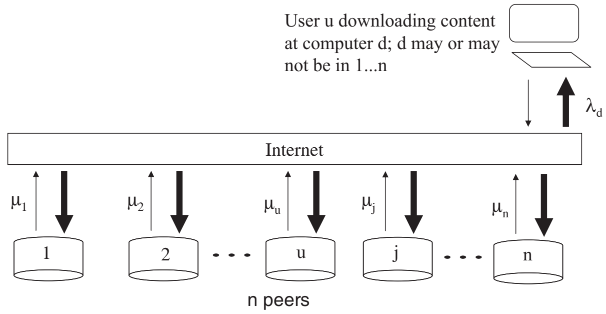


Fig. 2. The model of the proposed application. Peers are typically connected to the Internet by low bandwidth upload links and high bandwidth download links. User u owns peer u but downloads content from the peer network at some remote computer d, ideally at download rate more than μ_u.

We show a pictorial representation of one such network in Figure 2. It should be noted that a user *u* can always download content from his own peer *u* in parallel with any peer *j* that might have copies of the content. This is not the model in most p2p content sharing networks, simply because a peer would not download content from itself (which it already has). The subtle difference is an important distinction between other p2p networks and the proposed application because it allows for the *removal* of the 'non-dominant' condition of the analysis presented in (8): $\mu_k \le \sum_{i,i\neq k} \mu_i,\ \forall k = 1 \ldots n$. Removing the non-dominant condition also results in the removal of the symmetric barter or 'tit-for-tat' requirement for fairness. As such, the contribution of peer j to u might not equal the contribution of peer u to j; only both peer's overall contribution to the system is made equal through the bandwidth allocation rules presented in this work. Thus, as compared to Bittorrent, our application maintains a longer term memory of prior bandwidth contributions as described in Section IV. In addition, unlike Bittorrent, the number of seeds (peers with relevant blocks), does not increase during the lifetime of a download session in our application because the downloaded content is unique for each user.

Because peer data will be cached on many other (possibly untrustworthy) peers in our application, our application should use data encryption to protect peer data. Our prior work on this topic (26; 27) explained how the inherent encryption provided through random linear codes could be used in our scenario. In the rest of this paper we will assume that the data security aspects are addressed through these mechanisms will instead concentrate on distributed resource allocation schemes, fairness, and incentives for inducing cooperative behavior between peers. In particular, we extend our work in (26; 27) to include new (bandwidth) allocation rules that provide more generalized incentives to cooperate, as well as stronger pair-wise fairness. In addition we present a real implementation of the proposed system and characterize the performance of the proposed allocation schemes through this implementation. The Section (Section VI) on the implementation also discusses some practical aspects of implementing the distributed allocation schemes in a real system.

III. System details

We next describe some implementation details of our system. Throughout this description we assume that each user u corresponds to his peer u on the network (*e.g.,* his home computer).

A. Initialization

Our system is initialized with each peer disseminating its data among other peers using a random linear coding approach motivated by the work in (17) (which applies coding to p2p storage applications). For the purposes of our analysis, we assume that each peer has an infinite amount of disk space so that there is no utility cost for caching other peers' data. Our particular extension of random linear coding efficiently provides secrecy, authenticity and availability of the disseminated data, as we shall see later in this section.

Consider a long file X consisting of b bits to be disseminated in an n-peer network. In the standard random linear coding approach, X is split into k chunks $\{X_1, X_2, \ldots, X_k\}$ with each chunk mathematically represented as an m-element vector with components in a finite field $\mathbb{F}_q$ of size $q = 2^p$ for some p, *i.e.,* $X_j \in \mathbb{F}_{2^p}^m$ with $mpk = b$. This formulation effectively translates file X into k vector chunks. These vectors are, in turn, coded into nk message vectors $\{Y_1, Y_2, \ldots, Y_{nk}\}$ whose i-th component Y_i is computed as

$$Y_i = \sum_{j=1}^{k} \beta_{ij} \cdot X_j,\ \ i = 1 \ldots nk, \tag{1}$$

where each β_{ij} is randomly chosen from $\mathbb{F}_q$ using a cryptographically strong pseudo-random number generator (such as is provided in (19)) seeded with a cryptographic hash of i, and a secret key known only to the encoding peer. By choosing β's at random and appropriately tuning parameters k, p, and n, we can insure that the k-tuples $\beta_i = [\beta_{ij}, \quad j = 1 \ldots k]$ are almost surely linearly independent (18). A deterministic guarantee of linear independence can be provided through testing at the encoding peer.

Our encoding is similar to the encoding proposed in (15) for network coding-based multicast (14), with two important technical differences:

1) rather than transmitting β's as message headers, we use them as a secret key, as explained in (26; 27);
2) rather than having peers transferring linear combinations of their information to others on the network, peers transmit exactly what was uploaded to their storage area.

The first difference guarantees that no peer can decode a message stored on its system unless it correctly guesses the k-tuple β_i (and *knows* that the guess is correct). The second difference ensures that peers do not need to perform any computation when messages are requested from them; they simply forward what they have stored.

To complete the initialization phase, each plain text message ID i is appended to the Y_i's of (1) and these encoded messages are then uploaded to the n participating peers (up to k messages per peer), where they are stored.

This entire initialization phase is executed when free upload bandwidth is available or when new peers join the network. If peer u has low upload bandwidth and/or many files to share, then this initialization process may take a long time; however, the file contents are always still available directly from peer u (as in the standard client-server model), even during the initialization phase.

Some peers may choose to conserve storage space by storing $k' < k$ messages. In this case peers will not have all the information needed to decode a given file (*i.e.,* there would have to be other accessible peers with at least $k - k'$ messages to make up the deficit). We wish to emphasize that normally peers *will* download concurrently from many other peers in order to speed up data transfer even if any or all of the peers hold k blocks each.

B. Accessing data

To decode a file, a user first requests a total of k messages $Y_{i_1} \dots Y_{i_k}$ from multiple peers, preferably in parallel. Peer j transmits to the user at any rate up its available upload capacity μ_j; we will show in Section IV that peer j has a natural incentive to provide a fair amount capacity to user u, and that user u is guaranteed a certain favorable minimum capacity from the overall network regardless of peer j's specific behavior. In practice, the rate at which user u receives data from j will also be limited by the user's download capacity, TCP flow control and fragmentation, and various network-level fluctuations, although we ignore these effects for the purpose of our analysis in Section IV

Once user u has received all k requisite messages from the network, he broadcasts a *stop transmission* message to all contributing peers. At this point, the user can multiply the vector of received messages by the inverse of the appropriate square sub-matrix of the coefficient matrix $\beta = [\beta_{ij}]$ to retrieve the file X; the rows of the coefficient sub-matrix are determined from the IDs of each received message, as explained in Section III-A.

IV. Fairness and incentive to join and cooperate

In this section we introduce an analytical verification of our system, with particular attention to its *fairness* and the *incentive* it provides users to *join* and *cooperate* within the network. To this end, we study the steady-state behavior of two bandwidth allocation rules. Both rules are distributed in the sense that they require only local information be readily available at every peer, thus avoiding the complexity of building hierarchical networks and/or centralized resource brokers.

Our first allocation rule provides participating users with an *incentive to join* the network. The user is guaranteed to receive at least her *isolation* bandwidth from the network, on average, but potentially much higher *instantaneous* bandwidth. Furthermore, this provision is guaranteed in the presence of adversarial users colluding to gain more bandwidth from the network. We show that the allocation rule is *fair* in the sense that the users are allocated *free* network bandwidth proportionally to their contribution to the network.

Our second allocation rule provides a stronger symmetric barter of asymptotic average bandwidths, or a *tit-for-tat* relationship between every pair of users, when the number of independent users in the network grows large. Moreover, when the users' download demands are memoryless and most of the network is cooperative, then no other allocation rule can provide an individual or group of colluding users much better average download bandwidth. As such, minority groups have no incentive to seek alternative rules, thus providing users with *incentive to cooperate*.

Section outline In the next section we formalize our theoretical model, together with its definitions and assumptions. Thereafter, we motivate our specific approach through analysis of a similar system based on the *global proportional fairness* scheme of (8), which we show to provide incentive for user deception. Sections IV-B, and IV-C discuss our first allocation rule, proving its key properties: the incentive to join the network and fairness. Sections IV-E and IV-F focus on modifications that yield the second allocation scheme, which provides *pair-wise* fairness (*i.e.,* a tit-for-tat relationship) and incentive to cooperate, but requires each peer to accurately estimate average user demand. We complete our discussion in Section IV-F1 and IV-F2 with a method for estimating these average demands that asymptotically (in time) achieves the desired fairness properties. Finally, in Section IV-G we discuss practical implementation aspects of our approach, including the system's dynamic response time.

A. Theoretical Model

Our formal model considers n peers sharing their upload bandwidths in a time-slotted fashion, with peer i corresponding to a remote owner, user i. We further denote the individual upload bandwidth of peer i by μ_i and, for simplicity, we ignore the users' download capacities, assuming thus that the instantaneous download capacity of any user is larger than all the peers' upload bandwidths combined. We refer to the collection of all n peers and users as a *network* or *system* and assume that they are all interconnected and that the users' demands are being broadcast to all network peers.

We also assume an arbitrary demand pattern for each user, modeled with a binary indicator random variable $I_i(t)$ that is 1 if and only if user i requests bandwidth at time t. In the most general case, $\{I_i(t)\}_{t\geq 0}$ are dependent stochastic processes, which we term *demand* processes. In some parts of our analysis ,we will assume that the long-term time averages of these processes converge to

$$\gamma_i = \lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t} I_i(k),$$

and, furthermore, that the processes are ergodic, meaning that these time averages converge to their statistical expectations. Note that if user i operates *in isolation* and downloads only from its own peer, then its download speed is limited by μ_i per request, which corresponds to a long-term average capacity utilization of $\mu_i\gamma_i$ per time slot.

1) Allocation Rules: Our proposed bandwidth allocation schemes are provided as expressions for $\mu_{ij}(t)$, the upload bandwidth that peer i should devote to user j at slot t:

Allocation Rule 1

$$\mu_{ij}(t) = \frac{\mu_i}{\sum_{l=1}^{n} I_l(t)\sum_{k=0}^{t-1}\mu_{li}(k)} I_j(t)\sum_{k=0}^{t-1}\mu_{ji}(k), \tag{2}$$

Allocation Rule 2

$$\mu_{ij}(t) = \frac{\mu_i}{\sum_{l=1}^{n} \frac{I_l(t)}{\gamma_l}\sum_{k=0}^{t-1}\mu_{li}(k)} \cdot \frac{I_j(t)}{\gamma_j}\sum_{k=0}^{t-1}\mu_{ji}(k), \tag{3}$$

with some arbitrary small positive initial values for $\mu_{ji}(0)$. Note that $\mu_{ij}(t)$ is generally non-negative, and can be non-zero at time t only if $I_j(t) = 1$, that is user j has a request at that time.

Allocation Rule 2 requires each peer to have knowledge of all the users' average demands, γ_i, which in general is unavailable locally. In section IV-F1 we will discuss how to estimate γ_i locally at each peer while maintaining the network's asymptotic (in time) properties.

In contrast, Allocation Rule 1 relies solely on local measurements taken at each peer, and it does not require any transfer of information among the peers or users, a possible point of adversarial attack. In fact, this rule does not even require that the average demands converge, and we show within this general framework (*i.e.*, Theorem 1) that users are provided with an incentive to join the network. Under a stronger assumption that demand processes across different users are mutually independent, we can prove even stronger properties related to collusion attacks. This independence assumption is reasonable for a cooperative, large, and diversified network, as demands are generated by independent individual users.

In Section IV-F we provide an asymptotic analysis of Allocation Rule 2 under the assumption that the network involves a large number, n, of non-dominant users (8) and that the average demand and bandwidth of each user does not scale with n, meaning that $\mu_i, \gamma_i = \Theta(1)$. Adding a further assumption that demand processes are memoryless, meaning that $I_i(t)$ is independent of $\{I_i(k) : k < t\}$, we further show in Theorem 5 that, if most of the network is cooperating then no other allocation rule can guarantee a significantly larger average bandwidth for a single adversary or a small group of colluding users.

In the next section, we provide some motivation for our specific choices of allocation rules by analyzing the benefits and deficits of an existing sharing scheme.

B. Motivation

Let us consider first the case of independent user demands and an allocation scheme similar to the *global proportional fairness* scheme of (8). In this scheme the bandwidth allocated to peer i is proportional to peer i's contribution among all actively requesting peers at that time. That is,

$$\mu_{ij}(t) = \mu_i \frac{I_j(t)\mu_j}{\sum_{l=1}^{n} I_l(t)\mu_l}, \tag{4}$$

with the understanding that $0/0 = 0$. As a point of contrast, note that our allocation rules include self contributions μ_{ii} that are not part of the work in (8).

The instantaneous bandwidth that user j receives, $\sum_i \mu_{ij}(t)$, is at least μ_j, and typically larger since not all peers request bandwidth at all times (although they still contribute to potential requesters). For a lower bound it is convenient to rewrite equality (4) as

$$\mu_{ij}(t) = \mu_i \frac{I_j(t)\mu_j}{\mu_j + \sum_{l\neq j} I_l(t)\mu_l}, \tag{5}$$

since $\mu_{ij}(t) > 0$ only if $I_j(t) = 1$. Taking expectations, we get

$$\begin{aligned} E[\mu_{ij}(t)] &= \mu_i \gamma_j \mu_j E\left[\frac{1}{\mu_j + \sum_{l\neq j} I_l(t)\mu_l}\right] && (6)\\ &\geq \mu_i \frac{\gamma_j \mu_j}{\mu_j + \sum_{l\neq j} \gamma_l \mu_l}. && (7)\end{aligned}$$

Equality (6) holds since the numerator and the denominator of (5) are presumed independent, and inequality (7) is an application of Jensen's inequality (30). In particular,

$$E\left[\sum_i \mu_{ij}(t)\right] \;\geq\; \gamma_j\mu_j \frac{\sum_i \mu_i}{\mu_j + \sum_{l\neq j}\mu_l\gamma_l} \;\geq\; \gamma_j\mu_j,$$

where last inequality is strict unless $\gamma_l = 1$ for all $l \neq j$.

Fairness properties of the allocation rule of Equation (4) can perhaps be better understood when the network consists of many peers, each of which contributes a small amount of bandwidth. To develop some insight, let us assume for a moment that μ_j is $\Theta(1/n)$ whereas the per-user demand γ_j remains $\Theta(1)$ for all j. When the number of peers n is large the random sum $\sum_{l\neq j} \mu_l I_l(t)$ is roughly Gaussian (assuming independent demand processes), with mean $\sum_{l\neq j} \mu_l\gamma_l$ and variance $\sum_{l\neq j} \mu_l^2 \gamma_l(1-\gamma_l) = \Theta(1/n)$; therefore the lower bound in (7) becomes asymptotically exact. Furthermore, when $\mu_j = \Theta(1/n)$ the denominator in (5) is practically the same for all j; leading to

$$\mu_{ij}(t)\gamma_i \approx \mu_{ji}(t)\gamma_j. \tag{8}$$

Equation (8) implies that the network is *fair* in the sense that each user receives back the (normalized) amount of bandwidth its peer shares with any other user.

Despite these favorable properties, the allocation scheme (4) has at least one important drawback, as it lacks a mechanism for checks and balances and thus gives incentive to peers to misrepresent themselves. In particular

$$\frac{\partial}{\partial \mu_j}\left(\gamma_j\mu_j \frac{\sum_i \mu_i}{\mu_j + \sum_{l\neq j}\gamma_l\mu_l}\right) > 0,$$

providing incentive for peer j to declare (possibly deceptively) a high contribution μ_j. The problem could be avoided if peers could measure the actual overall contribution of *other* peers accurately. Alternatively, a peer can measure only contributions that it has received from each one of other peers, and it can then use this as a proxy for overall contribution in (4). This leads to the proposed Allocation Rule 1, which is studied in the next subsections.

C. Incentive and Fairness Analysis of Allocation Rule 1

We start by showing that Allocation Rule 1 provides an incentive to join the system, and, under certain reasonable conditions, this incentive is strong. We further show that our rule is pair-wise fair in the saturation region and that no other allocation rule can do better, under the same assumptions.

1) Definitions: We say that a user has an *incentive to join* a sharing system if he is guaranteed not to lose average bandwidth when he joins the network, assuming that the network users are cooperative in obeying a prescribed allocation rule. The user has a *strong* incentive to join if this guarantee holds even if the network users are not cooperative.

For our analysis, we will also use $\overline{\mu_{ij}}(t)$ to denote the time-average of the bandwidth that user j receives from peer i, and $\overline{\mu_j}(t)$ to denote the time-average over all bandwidth user j receives from the network, and $\overline{I_i}(t)$ to denote the time-average of the demand process:

$$\overline{\mu_{ij}}(t) = \frac{1}{t}\sum_{k=0}^{t-1}\mu_{ij}(k), \quad \overline{\mu_j}(t) = \sum_{i=1}^{n}\overline{\mu_{ij}}(t), \quad \overline{I_j}(t) = \frac{1}{t}\sum_{k=0}^{t-1} I_j(k). \tag{9}$$

Note that for all i and j, $0 \leq \overline{I_j}(t) \leq 1$ and $0 \leq \overline{\mu_{ij}}(t) \leq \mu_i$, so that the sequences $(\{\overline{I_j}(t), \overline{\mu_{ij}}(t),\ i,j = 1,2,\cdots,n\} : t \geq 1)$ lie in a compact Euclidean space. As such, for any sequence $\{t : t > 0\}$ there is a further subsequence $\{t_m : m \geq 1\}$ for which the time-averages sequence converge (31):

$$\overline{\mu_{ij}} = \lim_{m\to\infty}\overline{\mu_{ij}}(t_m), \quad \overline{\mu_i} = \lim_{m\to\infty}\overline{\mu_i}(t_m), \quad \gamma_j = \lim_{m\to\infty}\overline{I_j}(t_m).$$

The limit points $\overline{\mu_{ij}}$ and γ_j are, in general, functions of the sequence $\{t_m : m \geq 1\}$, but this is not an issue for our general result of the following theorem.

D. Incentive to join

Lemma 1 and Theorem 1 show that each user has a specific incentive to join our network.

Lemma 1 *Let $\{I_i(t) : t \geq 0\}, \quad i \leq n$, be arbitrary demand sequences. For any user i implementing Allocation Rule 1, the long-term average download bandwidth is related to the average bandwidth in isolation by,*

$$\overline{\mu_i} = \gamma_i \mu_i + \sum_l \overline{\mu_{li}} \left(1 - \frac{1}{\overline{\mu_{ii}}} \lim_{m\to\infty} \frac{1}{t_m} \sum_{k=0}^{t_m} I_l(k)\mu_{ii}(k) \right). \tag{10}$$

Proof: Rearranging (2) ,

$$\mu_{ij}(t) \left(\sum_{l=1}^{n} I_l(t)\overline{\mu_{li}}(t) \right) = \mu_i I_j(t)\overline{\mu_{ji}}(t). \tag{11}$$

Using the convergence property over the sequence $\{t_m \,|\, m \geq 1\}$, we fix a large enough m' so that we can substitute all $\overline{\mu_{ij}}(t_{m'})$ by $\overline{\mu_{ij}} + e_{ij}(t_{m'})$, and all $\overline{I_j}(t_{m'})$ by $\gamma_j + e_j(t_{m'})$, where the error terms $|e_j(t_{m'})|$ and $|e_{ij}(t_{m'})|$ are upper-bounded by some small $\epsilon > 0$, for all $m \geq m'$. Herein, we omit the error term subscripts (which do not affect the result) for sake of clarity.

Summing (11) from $t_{m'}$ to $t_m - 1$ and dividing by t_m gives

$$\frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} \mu_{ij}(t) \left(\sum_{l=1}^{n} I_l(t)(\overline{\mu_{li}} + e(t_{m'})) \right) = \frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} \mu_i I_j(t)(\overline{\mu_{ji}} + e(t_{m'})), \text{ for } m \gg m'. \tag{12}$$

For the right hand side term,

$$\frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} \mu_i I_j(t)(\overline{\mu_{ji}} + e(t_{m'})) = \mu_i \overline{\mu_{ji}} \frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} I_j(t) + \mu_i \frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} I_j(t) e(t_{m'}).$$

Noting that $\sum_{t=t_{m'}}^{t_m-1} I_j(t) = t_m \overline{I_j}(t_m) - t_{m'} \overline{I_j}(t_{m'})$ and substituting $\gamma_j + e(t_m)$ for $\overline{I_j}(t_m)$, gives

$$\mu_i \overline{\mu_{ji}} \left(\gamma_j + e(t_m) - \frac{t_{m'}}{t_m} \overline{I_j}(t_{m'}) \right) + \mu_i \frac{1}{t_m} \sum_{t=t_{m'}}^{t_m-1} I_j(t) e(t_{m'}) = \mu_i \overline{\mu_{ji}}(\gamma_j + e_1(t_m)) + e_2(t_m) = \mu_i \overline{\mu_{ji}} \gamma_j + e_3(t_m).$$

Since all the terms contributing to the errors are bounded, it follows that the total error term $e_3(t_m)$ can be made arbitrary small by increasing m. As such, the right hand side of (12) converges to $\mu_i \overline{\mu_{ji}} \gamma_j$ as m grows to infinity. An analysis of the left hand side similarly shows it to converge to $\frac{1}{t_m} \sum_{t=0}^{t_m} \mu_{ij}(t) \sum_l I_l(t) \overline{\mu_{li}}$.

The resulting equation, in asymptotic form, is thus:

$$\frac{1}{t_m} \sum_{t=0}^{t_m} \mu_{ij}(t) \sum_l I_l(t) \overline{\mu_{li}} \asymp \mu_i \gamma_j \overline{\mu_{ji}}.$$

Adding $\frac{1}{t_m} \sum_{t}^{t_m} \mu_{ij}(t) \sum_l (1 - I_l(t)) \overline{\mu_{li}}$ to both sides and recalling that $\sum_l \overline{\mu_{li}} = \overline{\mu_i}$, we get

$$\overline{\mu_{ij}}\ \overline{\mu_i} \asymp \mu_i \gamma_j \overline{\mu_{ji}} + \frac{1}{t_m} \sum_{t}^{t_m} \mu_{ij}(t) \sum_l (1 - I_l(t)) \overline{\mu_{li}}.$$

The result follows from considering the case $j = i$ case and dividing by $\overline{\mu_{ii}}$. ∎

Lemma 1 guarantees that all users receive at least the amount of bandwidth that they would have received in isolation. The following theorem shows that this result is tight in the sense that no better guarantee can be provided in the general case

Theorem 1 *Let $\{I_i(t) : t \geq 0\}, \quad i \leq n$, be arbitrary demand sequences then:*

1) *For any user i implementing Allocation Rule 1, $\overline{\mu_i} \geq \gamma_i \mu_i$, for all i (asymptotically in time), and*
2) *In fact, no other allocation rule can provide a higher guarantee (uniformly for all users) for arbitrary demand processes.*

Proof: To prove the first part of the theorem we observe that $\lim_{m\to\infty}\frac{1}{t_m}\sum_{k=0}^{t_m} I_l(k)\mu_{ii}(k) \leq \overline{\mu_{ii}}$, since the indicator function I is at most 1. As a result,

$$\sum_l \overline{\mu_{li}}\left(1 - \frac{1}{\overline{\mu_{ii}}}\lim_{m\to\infty}\frac{1}{t_m}\sum_{k=0}^{t_m} I_l(k)\mu_{ii}(k)\right) \geq 0,$$

and plugging it into equation (10) provides indeed that $\overline{\mu_i} \geq \gamma_i\mu_i$.

We prove the second part by contradiction. Assume that an allocation rule exists that guarantees every user i with strictly higher average bandwidth than its isolation bandwidth,*i.e.*, $\overline{\mu_i} > \gamma_i\mu_i$. If we sum this inequality over all users it follows that this allocation rule must guarantee that $\sum_i^n \overline{\mu_i} > \sum_i^n \gamma_i\mu_i$.

Next, consider a *synchronous* network, in which all demand requests are synchronized within a given time slot (*i.e.*, the demand sequences for all users are identical). In this degenerate situation the average bandwidth *available* to all users combined is equal to the sum of their isolation bandwidths; hence, regardless of the allocation rule $\sum_i^n \overline{\mu_i} \leq \sum_i^n \gamma_i\mu_i$, resulting in a contradiction. ∎

Note that Theorem 1 makes no assumptions about the nature of individual demand processes, and more importantly, its guarantees hold for any user implementing Allocation Rule 1, regardless of what other users might do. As such, even were the remaining network users to collaborate maliciously in their allocations, they could not (in the long term) reduce faithful users below their isolation bandwidth; this represents a strong incentive to join the network.

Furthermore, Lemma 1 provides that users following Allocation Rule 1 can expect a potentially positive gain beyond their isolation bandwidth, and this gain is roughly inversely proportional to the average amount of self contributions. Thus, our system is also *'fair'*, in the sense that users can expect to receive from the network in proportion to how much they give.

1) Independent demand processes: The specific case where demand processes are *independent across users* permits us to refine our fairness notion and gain more insight into our system. This case could reasonably appear for cooperative and diversified networks, where users are often anonymously sharing different types of resources.

From a technical perspective, our results in this section are predicated upon the assumption that the limit quantities $\overline{\mu_{ij}}$ are fixed for any subsequence t_m, or in other words that the sequences $(\mu_{ij}(t) : t \geq 0)$ converge. This is stated more formally in the following condition.

Condition 1 *For each i, j, the sequences $(I_j(t),\ \mu_{ij}(t) : t \geq 0)$ are asymptotically stationary and $\overline{\mu_{ij}}(t)$ converges as $t \to \infty$.*

Under this condition, we define

$$\overline{\mu_{ij}} = \lim_{t\to\infty}\overline{\mu_{ij}}(t), \quad \overline{\mu_j} = \lim_{t\to\infty}\overline{\mu_j}(t), \quad \gamma_j = \lim_{t\to\infty}\overline{I_j}(t) \tag{13}$$

to be the long-term averages corresponding to the quantities in (9)). Recall that an average upload bandwidth of peer i operating in isolation is $\gamma_i\mu_i$, and that the *unallocated* or *free* average bandwidth is $(1-\gamma_i)\mu_i$.

Theorem 2 *Let $\{I_i(t) : t \geq 0\}, \quad i \leq n$, be mutually independent demand processes of network users implementing Allocation Rule 1 and satisfying Condition 1. Then the average download bandwidth of user i is not only its average bandwidth in isolation but also fractional portions of the free bandwidth of other users in the network. That is,*

$$\overline{\mu_i} \geq \gamma_i\mu_i + \gamma_i\sum_{l\neq i}\alpha_{il}(1-\gamma_l)\mu_l.$$

where the fractional portions are proportional to the amount of the bandwidth user i shares with the network:

$$\alpha_{il} = \frac{\overline{\mu_{il}}}{\overline{\mu_{il}} + \sum_{j\neq i}\gamma_j\overline{\mu_{jl}}}.$$

Proof: Manipulating Allocation Rule 1 gives

$$\mu_{ij}(t) = \mu_i\frac{I_j(t)\overline{\mu_{ji}}(t)}{\overline{\mu_{ji}}(t) + \sum_{l\neq j} I_l(t)\overline{\mu_{li}}(t)}. \tag{14}$$

Under Condition 1this is well approximated (for large t) by

$$\mu_{ij}(t) = \mu_i\frac{I_j(t)\overline{\mu_{ji}}}{\overline{\mu_{ji}} + \sum_{l\neq j} I_l(t)\overline{\mu_{li}}}, \tag{15}$$

and, furthermore, the expectation of $\mu_{ij}(t)$ asymptotically approaches $\overline{\mu_{ij}}$. Since user demands are independent, the numerator and denominator of (15) are independent, and, taking expectations and applying Jensen's inequality, we get

$$\overline{\mu_{ij}} \;\geq\; \frac{\mu_i \gamma_j \overline{\mu_{ji}}}{\overline{\mu_{ji}} + \sum_{l \neq j} \gamma_l \overline{\mu_{li}}}. \tag{16}$$

For the case $i = j$ and adding $\sum_{l \neq i} \overline{\mu_{li}}$ to both sides, we get

$$\overline{\mu_i} = \overline{\mu_{ii}} + \sum_{l \neq i} \overline{\mu_{li}} \geq \gamma_i \mu_i + \sum_{l \neq i} (1 - \gamma_l) \overline{\mu_{li}}. \tag{17}$$

We complete the proof by expanding $\overline{\mu_{li}}$ using (16) in (17). ∎

Theorem 2 provides important fairness features of our allocation rule. First, it shows that each user has an incentive to share its bandwidth. The larger the bandwidth she shares relative to the amount others share, the larger the portion of free bandwidth that she receives. Second, a stronger *pair-wise fairness* condition holds in the saturated regime $\gamma_i \to 1$ for all i, meaning that, in this case, the average amount of bandwidth a pair of users share with each other is equal. A formal treatment of this property follows.

2) Fairness of the network in saturation: Thus far, we have shown that the network provides some guarantee on fairness by allocating larger portions of free bandwidth to users that provide more dominant portions to the upload of other users. We now show that in the saturated regime a strong *pair-wise fairness* holds (we will define this more generally in Section IV-E).

Corollary 1 *If $\gamma_i = 1$ for all i then the Allocation Rule 1 guarantees pair-wise fairness so that*

$$\overline{\mu_{ji}} = \overline{\mu_{ij}}, \qquad \forall i, j. \tag{18}$$

Proof: Inequality (16) reduces to

$$\overline{\mu_{ij}} = \frac{\mu_i \overline{\mu_{ji}}}{\overline{\mu_i}}. \tag{19}$$

in the considered regime, and holds for all i, j so that $\overline{\mu_{ji}} = \overline{\mu_{ij}}$. ∎

A similar result is given in (8), but our system permits self allocations μ_{ii} and thus no longer requires the *non-dominant* condition that each peer's upload bandwidth is less than the sum of all the other peers' upload bandwidths.

Note that the pair-wise fairness property of (18) does not hold in general. As previously indicated individual users can enjoy other peer's free upload bandwidth to increase their total average upload bandwidth even beyond their own single peer-user isolated bandwidth.

We next provide a modified allocation rule which seeks to provide fairness guarantees even when the network is not saturated, at the expense of additional assumptions.

E. Modifications to the Allocation Rule 1

Theorem 2 asymptotically provides a strong incentive for users to join our network. In this section we provide a motivation for a modified allocation rule that, under more general conditions than those of Section IV-D2, further provides *incentive to cooperate* as well as *pair-wise fairness*.

We start with formal definitions of these terms, with some similarities to the well-known Nash equilibrium.

Definition 1 *An allocation rule is said to be (ϵ, δ)-*fair *if, for all users and for arbitrary small, positive ϵ and δ, the average received bandwidth from the network is within ϵ of the average contributed bandwidth to the network (including self contributions) with probability at least $1 - \delta$:*

$$|\sum_l \overline{\mu_{li}} - \sum_l \overline{\mu_{il}}| < \epsilon \text{ with probability } 1 - \delta.$$

The allocation rule is also pair-wise fair *if* in addition

$$|\overline{\mu_{ij}} - \overline{\mu_{ji}}| < \epsilon \text{ with probability } 1 - \delta \text{ for all } i, j.$$

Definition 2 *A user u has an* incentive to cooperate *with a network of users implementing an allocation rule if, every positive ϵ and δ, no other allocation rule for u can provide an additional $1+\epsilon$ fraction of average download bandwidth with probability at least $1 - \delta$.*

1) Motivation: We first motivate and provide insight into our modified allocation rule with yet another hypothetical allocation rule, based on positive real numbers $\delta_1, .., \delta_n$ in the range $[0, 1]$.

$$\mu_{ij}(t) = \begin{cases} I_i(t)\mu_i \left[(1-\delta_i)\frac{\overline{\mu_{ii}}(t)}{\sum_l I_l(t)\overline{\mu_{li}}(t)} + \delta_i\right], & j = i \\ \\ \mu_i(1-\delta_i I_i(t))\frac{I_j(t)\overline{\mu_{ji}}(t)}{\sum_l I_l(t)\overline{\mu_{li}}(t)}, & j \neq i. \end{cases} \tag{20}$$

Allocation rule of Equation (20) guarantees that the i-th user gets at least $\delta_i\mu_i$ bandwidth from peer i each time it requests data. The remaining bandwidth is allocated according to the original Allocation Rule 1. For the degenerate case where $\delta_1 = \delta_2 = ... = \delta_n = 1$, each user is guaranteed its isolation bandwidth (namely $\gamma\mu$), for every time slot t, no matter how the rest of the network operates, and hence there is a strong incentive to join the network.

Consider the case of a small network, which has one user with high demand $\gamma = 1 - \epsilon$ for small $\epsilon > 0$. In the original allocation rule this user will allocate some bandwidth to its neighbors, which will entitle him to a large portion of the bandwidth of the network in the event that a small number (or none at all) users are requesting data. This asymmetry provides a high incentive for highly demanding users to join the system, exploiting its free bandwidth. Of course when only highly demanding users join the network the gain of each user is reduced and the system becomes ”fair” again, as indicated in Corollary 1.

On the other hand the modified allocation rule reduces the amount of bandwidth that a high demanding user shares with the network, hence reducing the probability that the bandwidth he shares becomes substantial compared to other, less demanding users. It follows that if each user is free to choose his own δ then, roughly speaking, highly demanding users would like to reduce their δ in order to gain more of the free bandwidth of the network, whereas other users would like to keep their δ high in order to guarantee their isolation bandwidth.

F. *Asymptotically (ϵ, δ)-pairwise fair Allocation Rule*

The observation that the Allocation Rule 1 tends to benefit highly demanding users, motivates us to consider a modification that simultaneously provides an incentive to join and cooperate *and* can be proven to be asymptotically (in the number of users) fair.

Allocation Rule 2 is similar to the original rule, with the exception that the time average of the received bandwidths is normalized by the average demands, with the usual convention of $\frac{0}{0} = 0$. For convenience, we rewrite the Allocation Rule 2 using long term averages notation:

$$\mu_{ij}(t) = \mu_i \frac{I_j(t)\frac{\overline{\mu_{ji}}(t)}{\gamma_j}}{\sum_l I_l(t)\frac{\overline{\mu_{li}}(t)}{\gamma_l}}. \tag{21}$$

Note that we explicitly assume that all average demands exist in the limit, and, furthermore, that these averages are locally known *a priori* to all peers. We demonstrate later how these averages can be efficiently estimated over time.

Theorem 3 *Let $\{I_i(t) : t \geq 0\}, \quad i \leq n$, be mutually independent demand processes satisfying Condition 1 under Allocation Rule 2. Then*

$$\overline{\mu_i} \geq \mu_i\gamma_i + (1-\gamma_i)\gamma_i \sum_{l \neq i} \mu_l \alpha_{il},$$

where $\alpha_{il} = \frac{\overline{\mu_{il}}}{\overline{\mu_{il}} + \gamma_i \sum_{j \neq l} \overline{\mu_{jl}}}$.

The proof is similar to the proof of Theorem 2 and is thus omitted.

Theorem 3 provides incentive to join the network but is weaker than Theorem 1 due to its additional assumptions on the demand processes. Nevertheless, we can extend it to prove pair-wise fairness (Theorem 4), when the number of users n grows asymptotically large.

For our benefit of analysis, we assume that the demand and bandwidth of every user i do not scale with n, more precisely $\mu_i, \gamma_i = \Theta(1)$ and, consequently, the network does not consist a *dominant peer*, whose bandwidth is larger than the sum of all the other peers (8).

The large number of users combined with the assumption of independent demands across users allows us to apply the strong law of large numbers to the sum

$$\sum_l I_l(t)\frac{\overline{\mu_{li}}(t)}{\gamma_l}.$$

This sum converges under Kolmogorov's criterion for independent *non-identically* distributed random variables (32), which requires the random variables to have finite variance, σ_i^2 and for the sum $\sum_i \frac{\sigma_i^2}{i^2}$ to converge. Clearly, these conditions hold in our case because $I_l(t)\frac{\overline{\mu_{li}}(t)}{\gamma_l} = \Theta(1)$.

Theorem 4 *Allocation Rule 2 is asymptotically (both in time and number of users n) (ϵ, δ) pair-wise fair for mutually independent demand processes satisfying Condition 1.*

Proof: Note that the Theorem does not specify the precedence of its asymptotics. We will prove the case when we take the limit over the number of users first. The other case follows similarly.

To prove the fairness property we observe that, given the average bandwidth allocations over time, the expectation of the denominator in the allocation rule is $\mathbf{E}\left\{\sum_l I_l(t)\frac{\overline{\mu_{li}}(t)}{\gamma_l}\right\} = \overline{\mu_i}(t)$. Assuming that there are $\Theta(n)$ non-zero allocations $\overline{\mu_{li}}(t)$, we can apply the strong law of large numbers, *i.e.*, for every $\epsilon, \delta > 0$ there exist n_0 such that

$$\forall n > n_0 \qquad |\overline{\mu_i}(t) - \sum_l I_l(t)\frac{\overline{\mu_{li}}(t)}{\gamma_l}| < \epsilon \text{ with probability at least} 1-\delta.$$

Taking expectations and relying upon Condition 1, the allocation rule results in

$$\overline{\mu_{ij}} \asymp \mu_i \frac{\overline{\mu_{ji}}}{\overline{\mu_i}}, \tag{22}$$

which, for $j = i$ gives that

$$|\overline{\mu_i} - \mu_i| < \epsilon. \tag{23}$$

Since every peer allocates all of its bandwidth when at least one user requests data, the average allocated bandwidth of the i-th peer is $\mu_i(1 - \prod_l 1 - \gamma_l) \to \mu_i$, and the (ϵ, δ) fairness of the allocation follows. Substituting equation (23) back into equation (22) proves the (ϵ, δ) pair-wise fairness property. ∎

Note that Theorem 4 provides a pairwise fairness in a more general setting than Corollary 1, since it does not require the network to be saturated. We next show that the modified allocation rule also provides an incentive to cooperate. Here we assume that the adversary's objective is to exploit as much of the network's free bandwidth as possible. Malicious users that try to disrupt but not game the network's operation are not considered in this analysis. Furthermore, we assume that the adversary cannot predict future user demands; though this assumption clearly holds for memoryless demand processes, it might be naive for many practical systems.

Theorem 5 *Allocation Rule 2 asymptotically (both in time and number of users n) provides an incentive to cooperate for mutually independent demand processes satisfying Condition 1.*

Proof: To see why the allocation rule provides an incentive to cooperate, consider a single user m who tries to maximize his long term average download bandwidth, while the other users cooperate with the allocation rule. We show that no strategy can guarantee a significant gain in the average download bandwidth.

Under Condition 1, the allocation rule can be rewritten (for $i \neq m$):

$$\mu_{ij}(t) = \mu_i \frac{\frac{I_j(t)}{\gamma_j}\overline{\mu_{ji}}}{\sum_{l \neq m} \frac{I_l(t)}{\gamma_l}\overline{\mu_{li}} + \frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}}$$

Using the law of large numbers, for an arbitrary $\epsilon' > 0$ there exists $n > n_0$ such that, almost surely,

$$\mu_i \frac{\frac{I_j(t)}{\gamma_j}\overline{\mu_{ji}}}{\sum_{l \neq m} \overline{\mu_{li}} + \epsilon' + \frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}} \leq \mu_{ij}(t) \leq \mu_i \frac{\frac{I_j(t)}{\gamma_j}\overline{\mu_{ji}}}{\sum_{l \neq m} \overline{\mu_{li}} - \epsilon' + \frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}}. \tag{24}$$

We first explore the left hand inequality, and we note that the denominator and numerator are independent under our assumptions. Assuming that the true demand of user m can be estimated accurately by the other users, then by taking the expectation and using Jensen's inequality, the left hand inequality becomes

$$\overline{\mu_{ij}} \geq \mu_i \frac{\overline{\mu_{ji}}}{\sum_{l \neq m} \overline{\mu_{li}} + \overline{\mu_{mi}} + \epsilon'} = \mu_i \frac{\overline{\mu_{ji}}}{\overline{\mu_i} + \epsilon'}.$$

and, in the case $i = j$, we have for all $i \neq m$,

$$\overline{\mu_i} \geq \mu_i - \epsilon'. \tag{25}$$

We next turn to the right hand inequality of (24) and consider the case $j = m$:

$$\mu_{im}(t) \leq \mu_i \frac{\frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}}{\sum_{l \neq m} \overline{\mu_{li}} - \epsilon' + \frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}} = \mu_i \frac{\frac{I_m(t)}{\gamma_m}\overline{\mu_{mi}}}{\sum_{l \neq m} \overline{\mu_{li}} - \epsilon' + \frac{1}{\gamma_m}\overline{\mu_{mi}}}.$$

Letting $\mu^* = \min_i \mu_i$ and $\epsilon^* = \frac{2\epsilon'}{\mu^*}$, the independence of numerator and denominator provides that

$$\overline{\mu_{im}} \leq \mu_i \frac{\overline{\mu_{mi}}}{\overline{\mu_i} - \epsilon' + \left(\frac{1}{\gamma_m} - 1\right)\overline{\mu_{mi}}} \leq^{(a)} \mu_i \frac{\overline{\mu_{mi}}}{\overline{\mu_i} - \epsilon'} \leq^{(b)} \mu_i \frac{\overline{\mu_{mi}}}{\mu_i - 2\epsilon'} \leq^{(c)} \frac{\overline{\mu_{mi}}}{1-\epsilon^*},$$

where $^{(a)}$ is due to the fact that $\frac{1}{\gamma_m} - 1 \geq 0$, $^{(b)}$ follows from (25), and $^{(c)}$ is true because we assume that $\mu_i = \Theta(1)$ for all i.

Finally, summing over all users i, and under our assumption that $\mu_m = \Theta(1)$, let $\mu_m^* \leq \mu_m$ be the expected bandwidth user m contributes to the network then for some arbitrary $\epsilon > 0$ there exist $n > n_0'$, such that

$$\overline{\mu_m} \leq \frac{\mu_m^*}{1-\epsilon^*} \leq \mu_m^* + \epsilon.$$

∎

It follows that, under the above assumptions, no matter what user m does he can not gain much more than his own contributed bandwidth. Combining this result with Theorem 4, we conclude that user m has no incentive to seek alternative strategies beyond Allocation Rule 2. This result can be extended to any finite group of colluding users, as long as their total bandwidth remains $\Theta(1)$.

1) Estimating Average Demands: In practice, the average demands of all users are typically unknown *a priori*. Still, it can be shown that with a simple estimation of the average demand (performed by each peer) Theorems 3,4 still hold, for large enough t.

Consider the following allocation rule (which requires only local measurements):

Allocation Rule 3

$$\mu_{ij}(t) = \mu_i \frac{I_j(t) \frac{\overline{\mu_{ji}}(t)}{\sum_{k=0}^{t-1} I_j(k)}}{\sum_l I_l(t) \frac{\overline{\mu_{li}}(t)}{\sum_{k=0}^{t-1} I_l(k)}}, \tag{26}$$

where we have replaced γ_j by the empirical average $\overline{I_j}(t) = \frac{1}{t}\sum_{k=0}^{t-1} I_j(k)$, with the typical convention that $\frac{0}{0} = 0$.

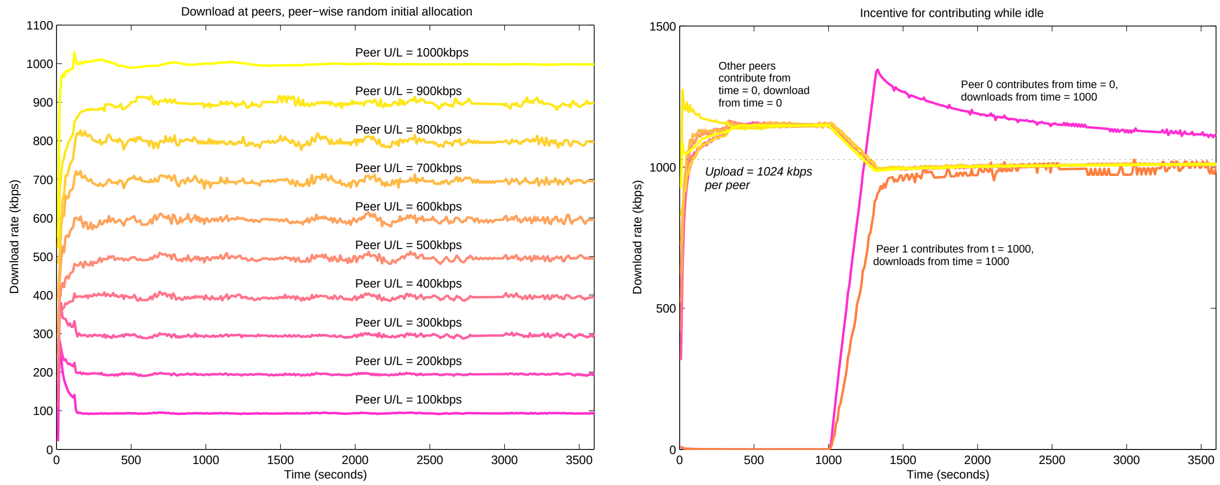
To see why Theorems 3 and 4 still hold (under Condition 1), we apply the strong law of large numbers: for large enough t, $\overline{I_j}(t) \asymp \gamma_j$. Hence equation (22) still holds, and the rest of the proof of Theorem 4 follows. The same reasoning applies also to Theorem 3.

2) Extensions: In the model discussed so far, the users' demands are assumed to be broadcast to all the peers in the network. One extension to our model would be to consider networks were the demands can be multi-casted to any desirable subset of peers. It can be shown that all the theorems except the incentive to cooperate (Theorem 5) still hold as long as the subset of peers stays fixed through the network's lifetime. Simply replace $I_i(t)$ with $I_{ij}(t)$, and γ_i with γ_{ij}, for every pair of users. Theorem 5 does not hold in this case, since the adversary can break the network into $\Theta(n)$ subnetworks in which the law of large numbers does not hold anymore.

G. Practical allocation rules

Unfortunately the allocation rules discussed need some tweaking for practical dynamic networks. Due to the long term averaging, changes in network capacity and demand with time are reflected very slowly in the sharing protocol. Consider, for example, the event of adding a new user to an existing network using Allocation Rule 1. This user will face a long "initialization" stage, during which it will contribute all its bandwidth to the system while accumulating enough "credit" to get its share of the network's free bandwidth. The situation is quite different when the distributed version of Allocation Rule 2 is used (Equation (26)). Here the lack of credit is canceled out with the lack of historic demands. In fact, the network will tend to welcome new users with a surge of free bandwidth much larger than their contributions. Although, this warm welcome may be interpreted as additional incentive to join the system, users may exploit this behavior to receive much more bandwidth from the system by bursting their demands while keeping their long term demand's average constant.

To reduce the surge of free bandwidth, we set the estimates of the initial demand γ to 1 (*i.e.,* $\sum_{k=0}^{t-1} I_j(k) = t$ for users joining the system at time t). Note that both Allocation Rules 2 and 1 become identical when $\gamma = 1$, so that when a new user enters the network it will initially build credit in as in Allocation Rule 1, but quickly adapt as the estimates of the average demands become more accurate. Clearly, the initial value for γ is irrelevant for the asymptotic (in time) properties of the network.



(a) Ten users request a large file from the system. Their download rate converges to the upload rate (U/L) of their peers.

(b) A peer that contributes while not using the network's resources gets rewarded later. A peer that joins later without contributing earlier suffers comparatively lower download rates.

Fig. 3. Convergence of the proposed approach and benefit of contributing bandwidth to the system.

V. Simulations and Experiments

We have implemented a discrete time simulator and an experimental prototype of the p2p system described in Section III for the purposes of demonstrating our claims. The results are presented in the following subsections, and are divided into two main groups. In the first group, we use our discrete time simulator to demonstrate the theoretical claims of Section IV. As such the simulator assumes a simplified traffic model, where user i requests bandwidth for download at time-slot t with probability γ_i, independently of other users and of the history of the system at slot t. In the second group of results, we present the effectiveness and implementation efficiency of random linear coding. Finally, we provide experimental results of our prototype system and simulation using a more realistic traffic model.

A. Allocation Rule 1

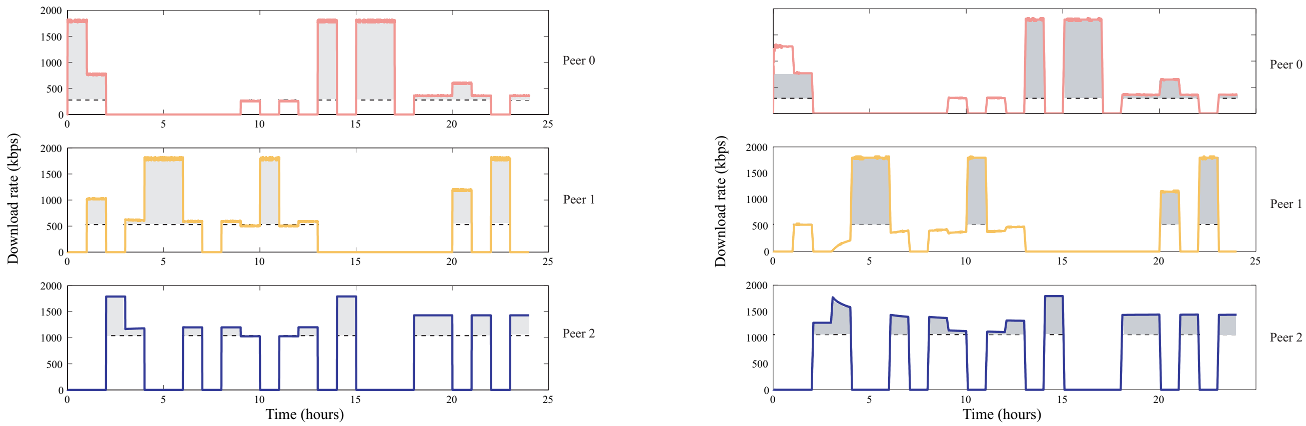
Our simulator permits nodes to initially allocate any feasible amount of upload bandwidth to their peers, although our specific experiments assume small, equal, non-zero initial contributions between peers. In our experiments, each peer reallocates its upload bandwidth once per second, and our graphs were smoothed over a running average of 10 seconds.

1) Fairness and benefit: In our first experiment, ten users request large files from the system at the same time, but the upload capacities of their corresponding peers range from $100kbps - 1000kbps$. Figure 3(a) shows the download rate available to the user of a peer as plotted against the time in seconds . Initially no one has downloaded any content from the other peers and the peers split their bandwidth randomly among all other requesting peers. All users demand bandwidth starting at time 0, although their corresponding peers are serving at different rates ($100kbps - 1000kbps$). The distributed system quickly converges to a fair allocation of users receiving what the (corresponding) peers contribute.

Our next experiment demonstrates that there is an incentive for peers to contribute even when their associated users are not downloading from the network, as illustrated in Figure 3(b). In this experiment, peer 0 steadily contributes bandwidth to the system, but peer 1 does not contribute bandwidth for the first 1000 seconds; neither peer 0 nor peer 1 request any files from the system during the initial 1000 seconds, and, thus, other peers take advantage of peer 0's unused bandwidth to get a download capacity that is greater than their upload capacity (*i.e.*, 1024 kbps). At time $t = 1000$ seconds, peer 1 starts contributing to the system and both peers 0 and 1 start requesting large files as well. We see that user 0 receives better service than user 1 because of the credited contribution of peer 0.

Our next few graphs demonstrate the benefits of the proposed system to all collaborating peers, commensurate with Theorem 2. For these graphs, we simulate a three peer network consisting of users that have encoded and distributed home videos to all three peers. The users stream their home videos to some remote computer for 12 randomly chosen hours in a day, meaning that users downloaded for half of the day in chunks of 1 hour.

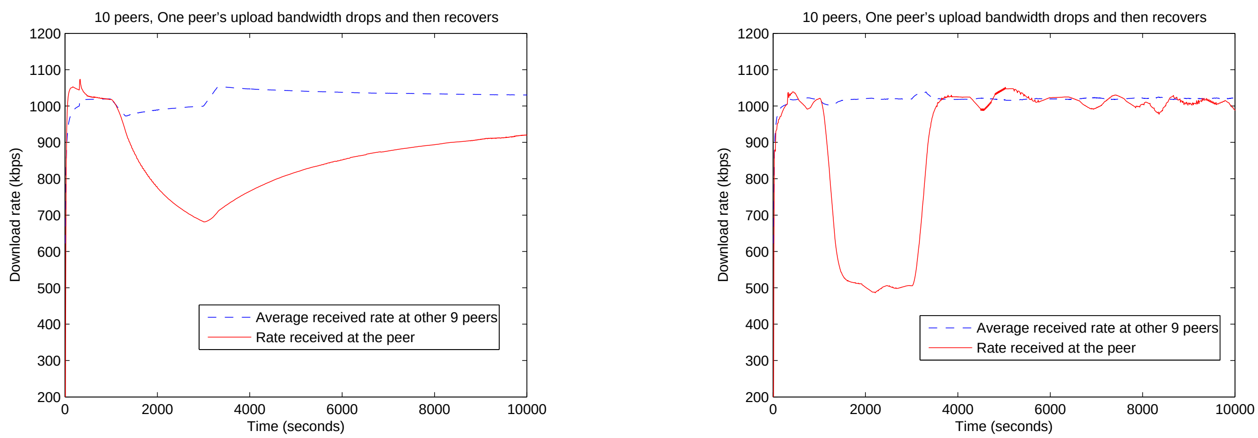
Figure 4(a) shows a case when peers 0, 1 and 2 have upload bandwidths $\mu_0 = 256kbps$, $\mu_1 = 512kbps$, $\mu_2 = 1024kbps$ respectively, and their duty-cycles correspond to the areas of non-zero download capacity. Each peer is available to upload to other peers throughout the 24 hour period, and we see that this cooperation benefits each user with a download capacity greater than he would receive in a single-user environment (shaded areas indicate gains).



(a) Each peer contributes all through the 24 hour period. The shaded regions are indicative of the gains of using Allocation Rule 1.

(b) Peer 1 starts contributing after the first 3 hours. The shaded regions are indicative of the gains of using the proposed approach.

Fig. 4. 3 peer network, $\mu_0 = 256kbps$, $\mu_1 = 512kbps$, $\mu_2 = 1024kbps$.



(a) Simulation scenarios showing incentive to contribute while idle, and the dynamics of Allocation rule 1

(b) Simulation scenarios showing incentive to contribute while idle, and the dynamics of a modified approach with "time-decaying" contributions of other peers. The denominator of Equation (2)) is multiplied by a decay factor 0.99 in every time-slot.

Fig. 5. Dynamics of Allocation rule 1

Figure 4(b) shows a case when peers 0, 1 and 2 have upload bandwidths $\mu_0 = 256kbps$, $\mu_1 = 512kbps$, $\mu_2 = 1024kbps$ respectively, and their duty-cycles correspond to the areas of non-zero download capacity. Each peer is available to upload to other peers all through the 24 hour period, and peer 1 only starts contributing to the system after the first three hours.

Two interesting artifacts occur: first, we notice that peer 1 is still able to get some service from the network in the first hour because peer 2 has not yet requested anything from the network and is splitting its bandwidth between peers 0 and 1, being oblivious to the fact that peer 1 is not contributing (this is corrected in the 2-3 hour time slot). The second artifact concerns the 3-4-hour time slot, in which peer 1 is penalized for his non-contribution to the system, though this penalty decays by time $t = 4$ hours, as all peers start benefiting from the contributed bandwidth. Note that although Peer 2 is a dominating peer ($\mu_2 > \mu_0 + \mu_1$), the system converges to a steady state.

In order to test the dynamics of the peer-wise proportional approach, we simulated a ten peer network with initial upload rates of 1024kbps per peer. All users request service throughout. At time $t = 1000$ seconds one peer's upload bandwidth contribution drops to 512kbps, and there is a consequent decrease in its download bandwidth, as shown in Figure 5(a). Interestingly, the other peers quickly recover the lost service amongst themselves. Then at time $t = 3000$ seconds, this peer's upload bandwidth contribution is restored to 1024kbps and the associated user's bandwidth is restored accordingly.

We should note that the system has slow dynamics, which could be speeded up by disproportionately weighing newer contributions over older ones. Employing an exponential time decaying weight function on bandwidth utility received by a

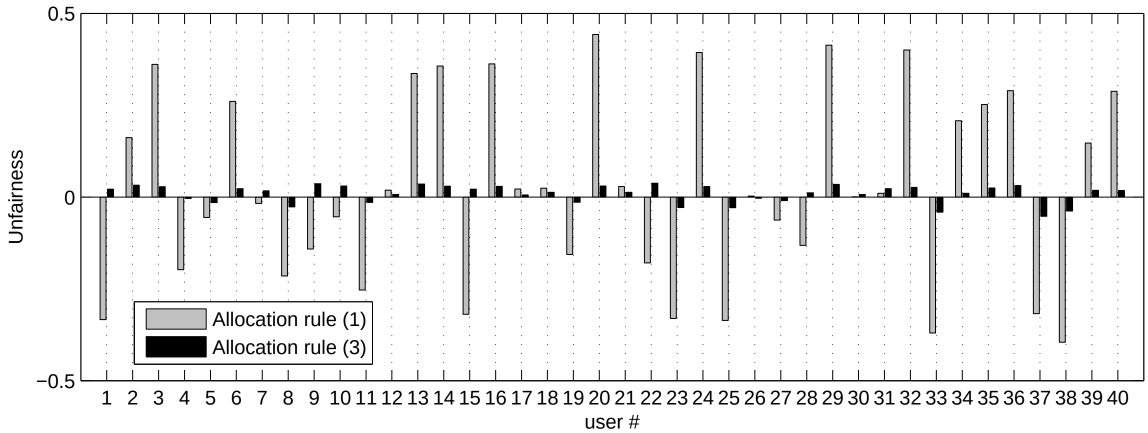


Fig. 6. $\frac{\overline{\mu_i}}{\mu i} - 1$ as unfairness measure for Allocation Rules 1, 2 with 40 users.

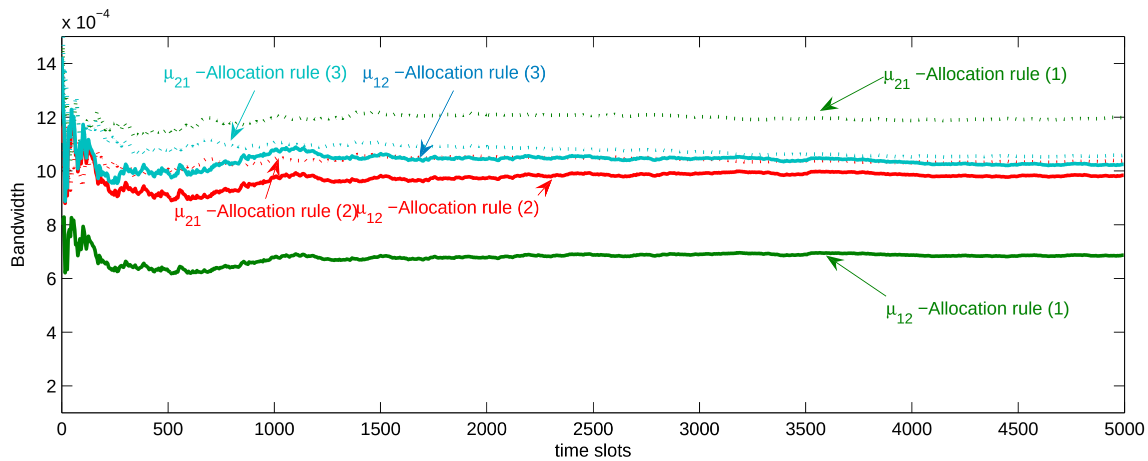


Fig. 7. $\overline{\mu_{12}}$ and $\overline{\mu_{21}}$ as a function of time for a simulation of 40 users, with different allocation rules.

peer will speed up the dynamics of the system (see for example, Figure 5(b)); we leave a thorough analysis of this decay for future work.

B. Allocation Rules 2 and 3

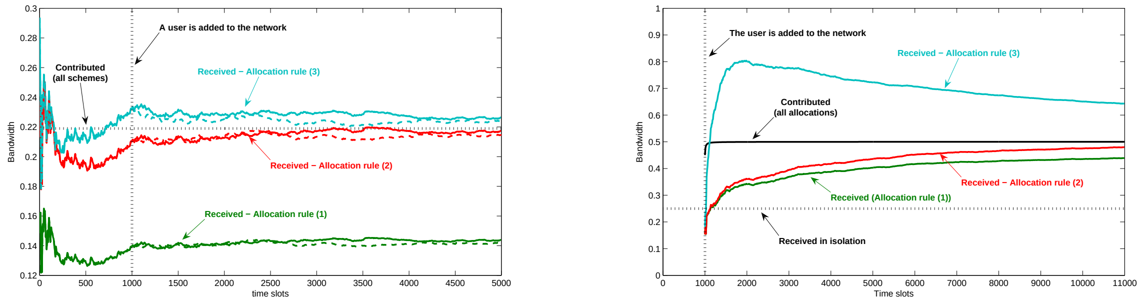
In this section we compare the different allocation rules for both static and dynamic scenarios. We focus here on asymptotic (ϵ, δ)- fairness and the effects of newly joint users. We assume a cooperative network, where the peers' capacities and average demands are arbitrary but normalized within range $[0, 1]$.

Figure 6 shows *unfairness* measure defined as the ratio between the average bandwidth received to the one contributed minus 1, namely $\frac{\overline{\mu_i}}{\mu i} - 1$. These measure should be close to 0 for all users in a fair network. The unfairness measures are calculated for a simulation of a network of 40 users with arbitrary capacities and average demands after $50,000$ time slots. Clearly, Allocation Rule 3 achieves better fairness results for all users.

Figure 7 shows the mutual average bandwidths of users 1 and 2 as a function of time. All allocation schemes have a similar convergence behavior, yet Allocation Rule 3 achieves better strong fairness results, as $\overline{\mu_{12}}$ and $\overline{\mu_{21}}$ tend to converge.

We next compare some of the dynamic properties of the different allocation rules. In the dynamic scenario, a user with both average demand and capacity of 0.5 is added to a network of 40 users after 1000 time slots. Figure 8 shows the behavior of the different allocation rules for the existing user (user 1) and for the new user. The impact of the added user on the existing users is small, as can be observed in Figure 8(a), where the average received bandwidths for both static (no user is added) and dynamic scenarios are shown. The differences between the allocation rules are evident in Figure 8(b), which shows the average bandwidths of the new user. Allocation Rule 3 provides considerable excess bandwidth, while Allocation Rules 1 and 2 show a monotonic convergence. It can also be observed that Allocation Rule 2 converges more rapidly to the contributed bandwidth as expected.

The encoding and decoding operations are essentially the same, the latter using the inverse of the coefficient matrix in Equation (1), so we only present decoding times in our results.



(a) Bandwidth as a function of time for user 1 for different allocation rules. The effect of the new user is shown by the dashed lines.

(b) Bandwidth as a function of time for the newly added user.

Fig. 8. Bandwidth as a function of time for different allocation rules. At time slot 1000 a new user is added to the network.

The number of messages k required to encode 1MB worth of data depends on different field sizes and message lengths. Thus, for example, if we used finite field of size $q = 2^{32}$ (*i.e.*, 32 bits/symbol) and a message length $m = 2^{15} = 32768$, then we would have $k = 8$ messages. This number is important because it defines the size of the β coefficient matrix. The number of field operations required to decode messages in Equation (1) is $O(mk^2 + mk)$. A very large k would also make the inversion of the coefficient matrix slow ($O(k^3)$ field operations), although in all our experiments the value of k was reasonably small and the matrix inversion time was negligible.

From Table I it is apparent that lower values of k yield faster decoding times. What is more important is that it makes sense to use larger field sizes to further reduce k, even with the additional overhead of more expensive field operations; however, reducing k indiscriminately would be problematic in maintaining fairness due to quantization errors. Table I shows that our example of $q = 2^{32}$ and $m = 32,768$ can be decoded at the rate of $1MB/s$. The bottleneck at this speed will probably be the ISP-offered download rate rather than the computations or upload capacities of the various peers.

VI. Implementation

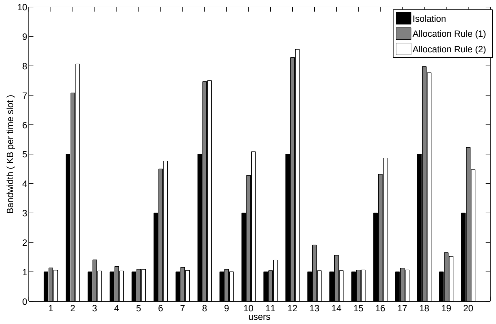
A prototype system was developed that combines random linear codes, a checksum using MD5 (33) at the user's end, and the allocation rule at the peer's end. The implementation is flexible, so that the peers could be distributed over physically different machines or grouped on one machine. Files are divided into 1 MB sub-files and then sub-divided into $\frac{1}{8}$ MB encoded chunks and stored on the peers. Users request each subfile individually from all peers. After all sub-files are downloaded they are decoded and an MD5 sum is computed to verify that decoding was successful. Any sub-files that did not decode properly due to linear dependence are re-requested. This implementation assumes that initialization (encoding and transferring of files to the peers) is completed before users request files. In practice this means that files cannot be added or modified after system start up without a complete reinitialization.

While our prototype implementation attempts to mirror the theoretical model, practical issues required small deviations. When peers allocate data, there is a minimum number of bytes that can be realistically allocated. This means that a user i that has contributed $\overline{\mu_{ij}} << \overline{\mu_j}$ may not receive bandwidth from user j. In this case user i is not being credited for all of its shared bandwidth and will receive less bandwidth from the network than it should in the theoretical model of Section IV-A. The theoretical model also assumes each user needs all bandwidth allocated by the network, but in practice this is not the case. If a user obtains enough data from the network to decode, then all remaining allocated bandwidth is superfluous and could be ignored (and, in fact, is not credited by the receiving peer).

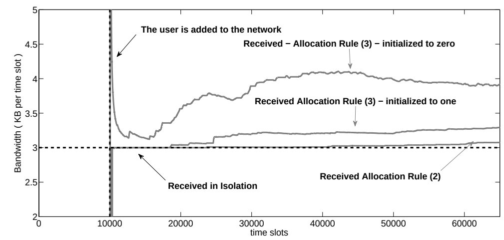
Our results were collected when all peers are located at one workstation, so that their total upload bandwidth is upper bounded by the workstation's upload bandwidth. The users were distributed over physically different terminals. The prototype implementation was initialized to simulate a network of remote clients requesting and downloading mp3 files, randomly chosen from a library of 3 MB files. The experiment had 20 peer/user pairs and ran for $200,000$ time slots. To simulate actual requests, users were configured to request files from the network according to a probability distribution fitted to data of inter-query times for a p2p network (28). This distribution represents the time between individual queries (search requests) on active connections for peak use in North America. The model is applicable to this experiment if it is assumed that each query represents a file request.

Figure 9(a) demonstrates that for both our schemes, every user gained more than its isolation bandwidth in each time slot that a file was demanded. This verifies the theoretical claim of our system providing a natural incentive to join.

In terms of fairness Allocation Rule 2 was found to be superior with 0.0963 *average unfairness* defined as $\frac{1}{n}\sum_{i=1}^{n}|\frac{\overline{\mu_i}}{\mu_i} - 1|$ compared to 0.13 average unfairness of Allocation Rule 1, in spite of the implementation deviations from the theoretical model as well as the inaccuracies in the estimations of the average demands.



(a) Average instantaneous bandwidth of Allocation Rules 1 and 2 compared to isolation bandwidths.



(b) Bandwidth of new user added at time slot 10,000 for different allocation rules.

Fig. 9. Prototype system simulations for different allocation rules.

$q \downarrow$, $m \rightarrow$	2^{14}	2^{15}	2^{16}	2^{17}
GF(2^4)	58.8	30.05	14.99	7.57
GF(2^8)	17.52	8.85	4.46	2.29
GF(2^{16})	5.53	2.81	1.42	0.72
GF(2^{32})	1.96	1	0.51	0.26

TABLE I
DECODING (ENCODING) TIMES IN SECONDS.

We also gathered preliminary data on the dynamic properties of our prototype implementation with Allocation Rules 2 and 3. In our dynamic scenario a new user (client and peer) are added to the network after $10,000$ time slots; the results are shown in Figure 9(b). Here the slow process of building credit is evident for the new user when Allocation Rule 1 is used. In contrast, if Allocation Rule 3 is used, the new user tends to receive a large amount of bandwidth from the network as its demand is estimated to be near zero. The middle ground is using Allocation Rule 3, with the estimated demands initialized to 1. This scheme removes the initial surge seen when a new user joins the network, but reduces the ”credit buildup” stage seen when Allocation Rule 3 is used.

We next demonstrate the efficiency of the random linear coding component of our system (and the corresponding decoding complexity). In order to establish the speed of random linear coding and infer the maximum throughput when the bottleneck is the decoding computation on the user’s computer, we have built a simple encoder/decoder following Equation (1) using Victor Shoup’s number theory library (19) in conjunction with the GNU multi-precision library (GMP) (20). We tested our system on 1MB of data for various values of message size m, finite field size q, and corresponding number of messages k into which the 1MB of data is split. The experiments were performed on a Pentium 4 dual processor workstation running the Linux operating system. For $m = 2^{32}$ and field size $q = GF(2^{32})$, decoding the 1MB (8Mb) data takes 1 second. At this speed the bottleneck will probably be the download link in broadband connection (typically less than 8Mb/s), rather than the decoding speed of random linear codes.

VII. CONCLUSIONS

In this paper we have discussed several distributed resource allocation schemes relevant to p2p systems, distributed grid computing, and other resource sharing scenarios. We have demonstrated the utility of our approach through a peer-to-peer application that enables users to overcome slow upload bandwidth bottlenecks when remotely accessing home server data. In our approach, several peers volunteer to disseminate the desired content, thus multiplexing their upload bandwidths in order to fill up the downloading user’s data pipe. This model fits very well with the typical user pattern of short periods of heavy link usage interspersed with long idle times.

To address issues of distributed access, our system stores information with the aid of random linear coding. This approach allows a user to reconstruct his file from a sufficient number of encoded packets, regardless of the source. We have also experimentally shown the computational feasibility of using random linear codes for large files. Our proposed approach provides a natural incentive for peers to voluntarily join and cooperate within our framework. Moreover, our system is asymptotically fair, in the sense that each user benefits from unallocated network bandwidth in proportion to its contribution to the system. As such, our system is also resilient to adversarial or malicious collusion, guaranteeing fairness even when some peers do not use the prescribed bandwidth allocation rule or attempt to interfere with others’ access. These analytic conclusions were individually confirmed through a variety of simulations, including a simulated home-video streaming network as well as experiments on our prototype implementation of the proposed system.

.

VIII. Acknowledgments

The authors wish to thank David Starobinski for editorial inputs.

References

[1] Foldershare.com, Bytetaxi, Inc.

[2] B. Cohen, “BitTorrent”, *http://bitconjurer.org/BitTorrent*.

[3] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao and J. Kubiatowicz, “Pond: The Oceanstore prototype”, *in Proc. of the USENIX Conference on File and Storage Technologies (FAST), 2004*.

[4] Joanna Glasner, “‘P2P fuels global bandwidth binge”, *http://www.wired.com /news/business/0,1367,67202,00.html*, *Wired*, April 14, 2005.

[5] D. Qiu, and R. Srikant, “Modelling and performance analysis of bittorrent-like peer-to-peer networks”, *in Proc. of ACM Sigcomm*, August 2004.

[6] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, “An analysis of Internet content delivery systems”, *in Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[7] S. Saroiu, P. Gummadi and S.Gribble, “A measurement Study of peer-to-peer file sharing systems”, *in Proc. of Multimedia Computing and Networking, 2002*.

[8] X. Yang and G. de Veciana, “Performance of peer-to-peer networks: service capacity and role of resource rharing policies”, *in Performance Evaluation - Special Issue on Performance Modeling and Evaluation of P2P Computing Systems*, 2005.

[9] R. Ma, S. Lee, D. Yau, “A game theoretic approach to provide incentive and service differentiation in P2P networks”, *in Proc. of ACE SIGMETRICS/Performance Evaluation Review*, 2004.

[10] T.S.E Ng, Y. Chu, S.G. Rao, K. Sripanidkulchai, and H. Zhang, “ Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems”, *in Proc. of IEEE Infocom*, April 2003.

[11] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose and Don Towsley, “Modeling peer-peer file sharing systems”, *in Proc. of IEEE Infocom*, April 2003.

[12] C. Gkantsidis, P. Rodriguez, “Network Coding for large scale content distribution”, *in Proc. of IEEE Infocom* March 2005, (Also as Microsoft Research Technical Report (MSR-TR-2004-80).

[13] L. Guo and S. Chen and Z. Xiao and E. Tan and X. Ding and X. Zhang, “Measurements, Analysis, and Modeling of BitTorrent-like Systems”,*ACM/SIGCOMM Internet Measurement Conference (IMC)*, 2005.

[14] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, “Network information flow”, *IEEE Trans. on Information Theory, vol. 46, pp. 1204-1216*, 2000.

[15] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding”, *In Proc. Allerton Conference on Communication, Control, and Computing*, October 2003.

[16] Eric W. Weisstein. “Coupon collector’s Problem”, *from MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/CouponCollectorsProblem.html*.

[17] S. Acedanski, S. Deb, M. Medard and R. Koetter, “How good is random linear coding based distributed Networked Storage?”, Invited talk, *http://www.netcod.org/*, *NetCod*, 2005.

[18] E. Calabi and H. S. Wilf. “On the sequential and random selection of subspaces over a finite field”, *Journal of Combinatorial Theory (A),* 22: 107109, 1977.

[19] V. Shoup,“Number theory library”, *http://shoup.net/ntl*.

[20] “GNU multi precision libary”, *http://www.swox.com/gmp/index.orig.html*.

[21] Q. Sun, H. Garcia-Molina, “SLIC: A selfish link based incentive mechanism for unstructured peer-to-peer networks”, *IEEE Proc. of the 24th Inter. Conf. on Distributed Computing Systems*, 2004.

[22] K. G. Anagnostakis, M. B. Greenwald, “Exchange-Based incentive mechanisms for peer-to-Peer file sharing”, *IEEE Proc. of the 24th Inter. Conf. on Distributed Computing Systems*, 2004.

[23] B. Yang, T. Condie, S. Kamvar, H. Garcia-Molina, “Non-cooperation in competitve P2P networks”, *IEEE Proc. of the 25th Inter. Conf. on Distributed Computing Systems*, 2005.

[24] N. Liogkas, R. Nelson, E. Kohler, L. Zhang“Exploiting BitTorrent for Fun (But Not Profit)”, *5th International Workshop on Peer-to-Peer Systems (IPTPS)*,2006.

[25] J. Song, C. Sha, H. Zhu, “Nash Equilibria in parallel downloading with multiple clients”, *IEEE Proc. of the 24th Inter. Conf. on Distributed Computing Systems*, 2004.

[26] S.Agarwal, M. Laifenfeld, A. Trachtenberg, and M. Alanyali, “Using bandwidth sharing to fairly overcome channel asymmetry”, *Information theory and applications workshop, UCSD*, 2006.

[27] S. Agarwal, M. Laifenfeld, A. Trachtenberg, and M. Alanyali, ”Fast Data Access over Assymmetric Channels Using Fair and Secure Bandwidth Sharing”, *International Conference on Distributed Computing Systems*, 2006.

[28] A. Klemm, C. Lindemann, M.K. Vernon, and O. P. Waldhorst,“ Characterizing the query behavior in peer-to-peer file sharing systems”, *IMC ’04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004.

[29] B. Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C”, John Wiley & Sons, Inc., 1996.

[30] T.M. Cover and J.A. Thomas, “Elements of Information Theory”, John Wiley & Sons, Inc., New York, 1991.

[31] W. Rudin, “Principles of Mathematical Analysis”, McGraw-Hill, 1976.

[32] W. Feller, “The Strong Law of Large Numbers”, *An Introduction to Probability Theory and Its Applications*, Vol. 1, 3rd ed. New York: Wiley, pp. 243-245, 1968.

[33] R.L. Rivest, “The MD5 Message Digest Algorithm”, *RFC 1321*, April 1992.

[34] T. Eymann, M. Reinicke, O. Ardaiz, P. Artigas, F. Freitag, L. Navarro, “Decentralized Resource Allocation in Application Layer Networks”,*CCGrid 2003: Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 645 - 650, May 2003.

[35] Wikipedia, “Grid computing”, *Wikipedia, The Free Encyclopedia*, 2008, *http://en.wikipedia.org/w/index.php*

[36] S. C. Lee, J. W. Jiang, D. C. Chiu, J. C. Lui,“Interaction of ISPs: Distributed Resource Allocation and Revenue Maximization”, *IEEE Trans. Parallel Distrib. Syst.* Vol. 19, Issue. 2, Feb. 2008, pp. 204-218.

[37] J. Yang, Y. Bai, T. Qiu,“A decentralized resource allocation policy in minigrid”,*Future Gener. Comput. Syst.*,Vol. 23, No. 3, pp. 359-366, 2007.

[38] C. Dixon, T. Bragin, A. Krishnamurthy, T. Anderson, “Tit-for-Tat Distributed Resource Allocation”, *ACM SIGCOMM ’06, Conference on Special Interest Group on Data Communication*, Sep 2006.