Fast data access over asymmetric channels using fair and secure bandwidth sharing

Sachin Agarwal
Deutsche Telekom Laboratories
Berlin, Germany-10587
sachin.agarwal@telekom.de
Moshe Laifenfeld Ari Trachtenberg Murat Alanyali
Boston University
Boston, MA, USA - 02215
{moshel, trachten, alanyali}@bu.edu

Abstract

We propose a peer-to-peer architecture designed to overcome asymmetries in upload/download speeds that are typical in end-user dialup, broadband and cellular wireless internet connections. Our approach allows users at remote locations to access information stored on their home computers at rates often exceeding their home connection's upload capacity. The key to this approach is to share file data when communications are idle using random linear coding, 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 prove that our proposed system is asymptotically fair, in that (even malicious) users are proportionally assigned idle bandwidth depending on how much bandwidth they contribute, and that there is a natural incentive to join and cooperate fairly in the system. In addition, our approach provides cryptographic security and geographic data robustness to the participating peers.

A version of this paper will appear as:

- S. Agarwal, M. Laifenfeld, A. Trachtenberg, and M. Alanyali, "Fast data access over asymmetric channels using fair and secure bandwidth sharing", Proceedings of the 26th International Conference on Distributed Computing Systems (ICDCS), July 4-7, 2006.

I. Introduction and Motivation

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 traditional modem dial-up, wireless cellular internet connections, and heterogeneous sensor networks.

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 when the biggest storage requirements for home users were typically from application programs or the operating system - static 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 channel utilization, as demonstrated by the example in Figure 1, where typical cable modem channel asymmetries lead to an access time of 9 hours for a one-hour MPEG-2 home video rather than 45 minutes for a completely utilized download link.

This work attempts to correct for such channel asymmetries by making it feasible to fill a high bandwidth download pipe with the aggregated services of several lower-bandwidth upload pipes that are otherwise idle. Our proposed approach has the following specific benefits:

Fairness: Unallocated bandwidth in the system is redistributed in proportion to the bandwidth contributed by each peer to the system.

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

Security: Shared data is secure and can be authenticated on the fly.

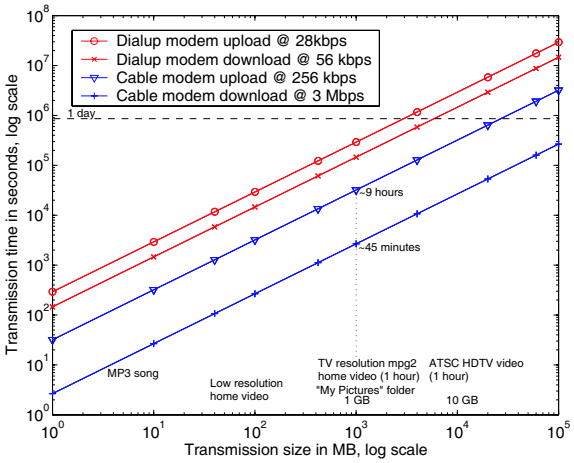


Fig. 1. Examples of asymmetry in upload and download capacities of connections that lead to significant differences between upload and download times for the same data; the difference sometimes spanning an order of magnitude.

Distributed operation: Only local information is needed in the implementation of the algorithm (i.e. no control information needs to be exchanged).

Robustness: Data is redundantly available from various sources.

In effect, our approach permits users to bypass the 'bandwidth: use it or lose it' service model offered by commercial ISPs, and instead maintain 'credit' for their contributions within the system as a whole. With hard-disk storage costing under a dollar per gigabyte, the benefits enumerated above quickly surpass the cost of caching other users' data. The rest of this paper is organized as follows. In Section II we mention some of the related work in the field and contrast it with our approach in Section II-A. Thereafter, we formally introduce the details of our proposed sharing method, together with its robustness and security properties, in Section III. In Section IV we analytically prove the fairness of our system 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 take unfair advantage of the system. We also implement and study the performance of random linear codes in order to demonstrate their real-time decoding practicability for this application. Finally Section VI concludes our results and suggests directions for future work.

II. Related work

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 [10]. 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 [20], [22], [18], [19], [17], [16], [15], [21], [23] 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 work 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 readily available to each peer. Similarly 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 [4], [38], [39], [40]. Although adding incentive scheme may increase the cooperation among reasonable users, namely users which try to optimize their resources, it usually has no guarantee against malicious peers and additional measures are required to reduce their effect on the network.

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 [9], provides a large internet-scale data storage solution for data security and reliability using erasure codes.

Various distributed file systems such as CODA [7] and Bayou [8], among others, have also been proposed for distributed storage and synchronization of information. The focus in these works is on issues related to file and data consistency and replication rather than solving the asymmetric bandwidth problem of end users.

Security The PASIS system [26] provides data security in addition to geographical redundancy for user data by means of a form of secret sharing [29], the emphasis being on reliable, authenticated and secure storage. The Eternity service [30] and the

Publius system [31] provide internet-scale distributed methods of secure and authenticated information storage and retrieval, although the focus in these projects is on the cryptographic aspects of distributed storage and user anonymity.

Coding Random linear coding [28], [33] has been used for achieving the network coding [27] min-cut bound on multicast in networks. The authors in [33] proposed random linear coding as a way to avoid the coupon collector's problem [32] 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, the 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. [38], [39], [40]). Nevertheless, there has been some recent queuing theoretic analytical work [22] 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 [41] 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 systems 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 [18], [19], *i.e.,* that the upload capacity of one 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).

The 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. In theory the upper bound on the download bandwidth available to peers is $\sum_{i=1\ldots n} \mu_i$, in practice this rate of downloading a file is upper bounded by the download bandwidth of the the particulars user's connection.

The idea is to fill up the thicker download pipe by employing the services of multiple narrow upload pipes. In most commercial asymmetric ISP connections the ratio between download and upload bandwidth is presently of the order of 5 to 10. This means that the proposed system will be useful as a small networks of 10s of nodes, making it small network relative to many P2P networks mentioned in Section II. A smaller network means that users can easily keep track of IP addresses of other peers through a simple out of band technique such as dynamic DNS services. Off course hosts are free to be part of more than one such bandwidth sharing network.

We show a pictorial representation of one such network in Figure III-A. It is to be noted that when a user *u* wants to download content while at host *d*, it can also include peer *u* in the list of peers it uses to download the content. This is not the model in most P2P content sharing networks, simply because a peer would not download content from itself that 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 [18]:

$$\mu_k \leq \sum_{i,i\neq k} \mu_i, \ \forall k = 1 \ldots n \tag{1}$$

Removing the non-dominant condition also results in the removal of the symmetric or 'tit-for-tat' requirement for fairness. In fact in the proposed application might not necessarily receive as much service as it contributes to another peer, but it does receive as much service as it contributes to the system.

Peer data will be cached on many other hosts in the proposed peer network. We present a solution to the problem of content security when personal content is stored on multiple untrustworthy peers without any additional overhead of encryption by using random linear coding as our free encryption mechanism, as explained in Section III-C.

III. System details

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

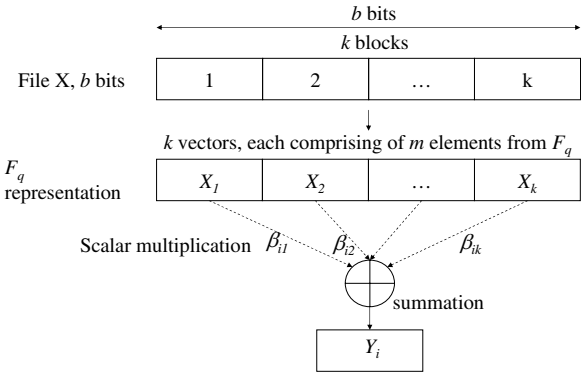


Fig. 2. The encoding process sums random linear combinations of the k input file vectors to produce each m-ary message vector Y_i in Equation (2).

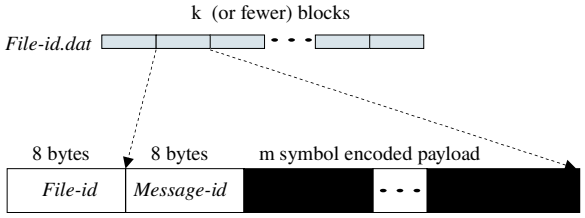


Fig. 3. Data file stored on a peer is composed of many 'pre-fabricated' messages that are transmitted from the peer serially to the downloading user.

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 [33] (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 another peer's data. Our particular extension of random linear coding specifically ensures the cryptographic secrecy, authenticity and availability of the disseminated data, as we shall see later in this section.

More precisely, consider a long file X consisting of b bits to be disseminated in an n-peer network, as diagrammed by Figure 2. 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

$$Y_i = \sum_{j=1}^{k} \beta_{ij} \cdot X_j, \;\; i = 1 \ldots nk, \tag{2}$$

where each β_{ij} is randomly chosen from $\mathbb{F}_q$ using a cryptographically strong random number generator (such as is provided in [36]) 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 [34]. A deterministic guarantee of of linear independence can be provided through testing at the encoding peer.

Our encoding is similar to the encoding proposed in [28] for network coding-based multicast [27], with two important technical differences: (1) rather than transmitting β's as message headers, we use them as a secret key; (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 (2) and these encoded messages are then uploaded to the n participating peers (up to k messages per peer) where they are stored as illustrated in Figure 3. In our case, we also compute a 128-bit MD5 hash of each uploaded message that the user stores and uses for message authentication as explained in Section III-B.

This entire initialization phase is executed when some 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 process can take a long time; however, the file contents are always still available directly from peer u (as in the standard client-server model) during the initialization phase.

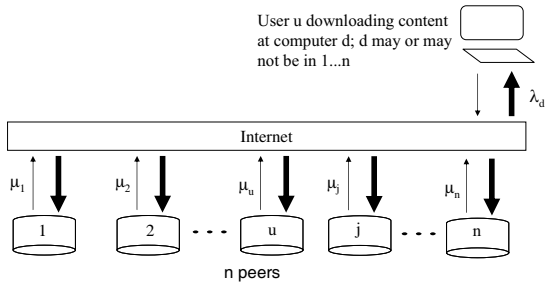


Fig. 4. The model of the proposed application. Peers are typically connected to the internet by low bandwidth upload links and high bandwidth download links. User u owns peer u but downloads content from the peer network at some remote computer d, ideally at download rate more than μ_u

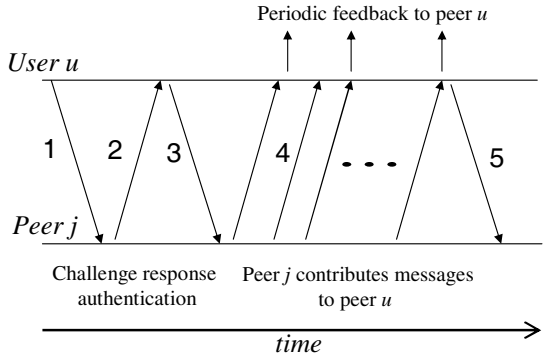


Fig. 5. System details

B. Accessing data

To decode a file, a user requests a total of k messages (*i.e.,* Y_i's) from multiple peers, preferably in parallel, and multiplies these messages by the inverse of the appropriate square sub-matrix of the coefficient matrix $\beta = [\beta_{ij}]$. The rows of the coefficient sub-matrix are determined from the message-id's appended to each of the k received messages Y_i (as in Section III-A).

A user u downloads data on some remote peer d, which has download bandwidth λ_d, as shown in Figure III-A. User u would typically contact multiple peers and request encoded messages comprising the desired (encoded) file. The time-line for a connection between user u (on computer d) and some peer j is depicted in Figure III-A. The transmission labeled 1 in the figure corresponds to an authentication of user u to j using a classic public-key challenge response system. Ideally, this authentication should go both ways (*i.e.,* peer j should authenticate to user u as well) in order to prevent man-in-the-middle or IP spoofing attacks.

Consider a user u connecting to a peer j to download data chunks of file X to user u. Peer j may choose to transmit to u at any rate up to its available upload capacity μ_j, although we show in Section IV that u can guarantee a certain download capacity from the peer network regardless of j's transmission rate. As mentioned in the introduction, in this paper we assume that the download rate constraint λ_d is too high to become a bottleneck in any of the file transmissions; this leaves the actual flow control to TCP, although it might be counterproductive to download content from too many peers due to excessive fragmentation of the download link at the user's end.

User u will typically contact multiple peers in order to fully use its download bandwidth λ_d. When enough encoded messages have been received to decode the complete file X, user u sends a stop transmission to contributing peers and reconstructs the file. User u also contacts its own peer u periodically with informational updates in order to let peer u make informed decisions on dividing its upload capacity among other users. This step can be done off-line, meaning that peer u does not have to be available for these purposes when user u downloads files.

C. Security

As mentioned earlier, the only way a peer can decode another user's encoded message is by correctly guessing the user's appropriate k-tuple of random coefficients β_i and knowing that this guess is correct (there is no generic way of verifying a guess). Even were a malicious peer to decode a message Y_i, this would not help in decoding the file unless $k-1$ other messages were also decoded. As such, the security of our system is tied to the security of the cryptographic hash (and secret) used for encoding.

Instead of decoding a message, a malicious peer could choose to inject fake messages into the network. For this reason, our system employs message-based authentication, storing 128-bit MD5 hashes for each message on the originating peer. For our

example cases in this paper we will utilize parameters $k = 8$, $m = 32,768$ and $q = 2^{32}$, corresponding to an authentication overhead of 128 hash bytes per megabyte; this overhead needs to be stored on (and retrieved from) a file's owner before downloading the file. If the owning peer is off-line, this information needs to be carried by the user.

D. Implementation considerations

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 the of the peers hold k blocks each.

For computational efficiency we wish to limit the number of messages k needed to reconstruct a given file. At the same time we also wish to avoid large message sizes m, which dilute our notion of fairness (see Section IV) by introducing quantization errors when nodes divide up their upload bandwidth amongst requesting users. We propose to overcome this problem by dividing large files into 1 MB chunks and then encoding each chunk as a separate file. The user will thus have to keep additional information about how such 1MB files fit together into a large file. However, this approach allows large files (*e.g.,* audio or visual data) to be 'streamed' to a user in small chunks, rather than forcing the user to wait until the entire file contents have been downloaded.

IV. Fairness and incentive to cooperate

In this section we introduce a technical analysis of our system. In particular we focus on the *incentive* and *fairness* aspects for peers comprising the network. In subsection IV-C we show that each user has an incentive to join and cooperate with the network. A user has an incentive to join the system since he is guaranteed to receive, on average, at least as much bandwidth as he would get had he had operated with his corresponding peer in isolation, but potentially higher instantaneous bandwidths. A user has incentive to cooperate since the amount of additional allocated bandwidth it receives from the system is proportional to the amount of bandwidth it allocates to the network. We will further argue that the system is resilient to adversarial users, that is, the incentive remains for any strategy adopted by one or several users. We will also show that the system is *fair* in the sense that the users are allocated *free* bandwidth according to how much bandwidth they share with the network (relative to others). We also show that when the network is saturated, *i.e.,* when all users constantly request data, the systems is *pair-wise fair*, namely every pair of users share the same amount of bandwidth.

We start in the next subsection with technical definitions and a formulation of our proposed system. As a motivation we study the properties of a different system in subsection IV-B.

A. Technical definitions

Our formal model considers n peers sharing their upload bandwidths in a time-slotted fashion. As indicated in previous sections, we will refer to the remote owner of a peer, say peer i, as user i. We denote the individual upload bandwidth of peer i with μ_i, and assume a random demand pattern for each user. Specifically, we assume that 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. Let $I_i(t)$ be the binary random variable that is 1 if and only if user i requests bandwidth at time t. The independence assumption above implies that the variables $I_i = (I_i(t) : t \geq 0)$ are independent identically distributed (iid) with $Pr(I_i(t) = 1) = \gamma_i$, and that the sequences $(I_i : i = 1, 2, \cdots, n)$ are independent.

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 an average capacity utilization of $\mu_i\gamma_i$ per time slot in the long term. We next introduce our bandwidth allocation scheme and study its properties. As such, let $\mu_{ij}(t)$ denote the upload bandwidth that peer i devotes to user j at slot t. This non-negative quantity is non-zero only if $I_j(t) = 1$, that is, user j has a request at that time. The proposed bandwidth allocation scheme is thus given by,

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

with some arbitrary small positive initial values for $\mu_{ji}(0)$. Note that the proposed scheme relies solely on local measurements taken at each peer, and it does not require any transfer of information among the peers or users, a point otherwise prone to adversary actions. As a motivation, we introduce a different allocation scheme in the next subsection and study its benefits and shortcomings.

B. Motivation

Let us consider first a similar allocation scheme to the *global proportional fairness* scheme of [18]. 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 $0/0 = 0$ (in contrast to [18], [19] our allocation scheme (4) includes self contributions, μ_{ii}).

The instantaneous bandwidth that user j enjoys, $\sum_i \mu_{ij}(t)$, is at least μ_j, and typically larger since not all peers request bandwidth (though they contribute to potential requesters) at all times. For a lower bound it is convenient to express equality (4) in the equivalent form

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

which follows since $\mu_{ij}(t) > 0$ only if $I_j(t) = 1$. Thus

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

Here equality (6) holds since the numerator and the denominator are independent in (5), and inequality (7) is an application of Jensen's inequality. 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 (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 $O(1/n)$ whereas the per-user demand γ_j remains $O(1)$ 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, with mean $\sum_{l\neq j}\mu_l\gamma_l$ and variance $\sum_{l\neq j}\mu_l^2\gamma_l(1-\gamma_l) = O(1/n)$; therefore the lower bound in (7) becomes asymptotically exact. Furthermore, when $\mu_j = O(1/n)$ the denominator in (5) is practically the same for all j; leading to

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

Note that (8) implies that the network is *fair* in the sense that each user receives back the (normalized) amount of bandwidth he shares with others.

Despite these favorable properties, the allocation scheme (4) has major drawbacks, primarily because it lacks a mechanism for checks and balances, and gives incentive to peers to misrepresent themselves. In particular

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

which can be interpreted as a strong 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. While this approach has clear implementational difficulties, a peer can easily measure the contribution that it has received from each one of other peers, and can use this as a proxy to overall contribution in (4). This leads to the proposed allocation rule of (3), which is studied in the next subsections.

C. Incentive and Fairness Analysis

Let $\overline{\mu_{ij}}(t)$ denote the time-average of the bandwidth that user j receives from peer i, and let $\overline{\mu_j}(t)$ be the time-average of the total bandwidth user j receives from the network, over the first t time slots. Namely,

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

Note that the sequence $(\{\overline{\mu_{ij}}(t),\ i,j = 1,2,\cdots,n\} : t \geq 1)$ is a Markov chain. Although transition probabilities of this chain are explicitly known, a rigorous equilibrium analysis entails techniques for time-inhomogeneous chains with non-discrete state space. Here we defer such technicalities by providing results under the following condition:

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

Under this condition let

$$\overline{\mu_{ij}} = \lim_{t\to\infty} \overline{\mu_{ij}}(t), \quad \overline{\mu_j} = \lim_{t\to\infty} \overline{\mu_j}(t) \tag{10}$$

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

Theorem 1 *Under Condition 1 the allocation rule* (3) *guarantees (asymptotically) that 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, i.e.,

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

Proof: We sketch the proof out of space considerations. Multiplying and dividing (3) by t we get

$$\mu_{ij}(t) = \mu_i \frac{I_j(t)\overline{\mu_{ji}}(t)}{\overline{\mu_{ji}}(t) + \sum_{l\neq j} I_l(t)\overline{\mu_{li}}(t)}. \tag{11}$$

In view of Condition 1 this allocation rule is well approximated for large values of t by

$$\mu_{ij}(t) = \mu_i \frac{I_j(t)\overline{\mu_{ji}}}{\overline{\mu_{ji}} + \sum_{l\neq j} I_l(t)\overline{\mu_{li}}}, \tag{12}$$

and furthermore the expectation of $\mu_{ij}(t)$ is asymptotically equal to $\overline{\mu_{ij}}$ as $t\to\infty$. Taking expectations in (12) and applying Jensen's inequality results in

$$\overline{\mu_{ij}} \;\geq\; \frac{\mu_i\gamma_j\overline{\mu_{ji}}}{\overline{\mu_{ji}} + \sum_{l\neq j} \gamma_l\overline{\mu_{li}}}. \tag{13}$$

Specializing to the case $i=j$ we get (after manipulations)

$$\overline{\mu_{ii}} \geq \gamma_i\mu_i - \sum_{l\neq i} \gamma_l\overline{\mu_{li}}. \tag{14}$$

By adding $\sum_{l\neq i}\overline{\mu_{li}}$ to both sides,

$$\overline{\mu_i} = \overline{\mu_{ii}} + \sum_{l\neq i}\overline{\mu_{li}} \geq \gamma_i\mu_i + \sum_{l\neq i}(1-\gamma_l)\overline{\mu_{li}}. \tag{15}$$

We complete the proof by using (13) to substitute for $\overline{\mu_{li}}$ ∎

Theorem 1 provides three important features of the allocation scheme. Firstly, each user has an incentive to share its bandwidth. The larger the bandwidth he shares relative to the amount others share, the larger the portion of free bandwidth that he receives. This can also be interpreted in terms of *fairness*: a more dominant peer (*i.e.,* one who has a larger portion in the upload bandwidth of user i) will benefit from a larger portion of the free bandwidth of this user.

The second important feature is the incentive to join the network. Theorem 1 guarantees that all users receive at least the amount of bandwidth that they would have received in isolation. Note that this guarantee for a specific user, say user i, holds under the mere assumption that this user requests downloads independently of the remaining users. Consider, for example, a set of users forming a coalition in order to manipulate the bandwidth of peer i. No matter what strategy they apply the guarantee of Theorem 1 still holds, implying that a user can only gain by joining the network.

Finally, a stronger *pair-wise fairness* condition holds in the saturated regime $\gamma_i \to 1$ for all i. Namely, the average amount of bandwidth a pair of users share is equal. A formal statement is the following corollary to Theorem 1:

In the next subsection we discuss further the fairness of the system in the stationary regime of $\gamma_i \to 1$ (*i.e.,* all peers are saturated with upload requests).

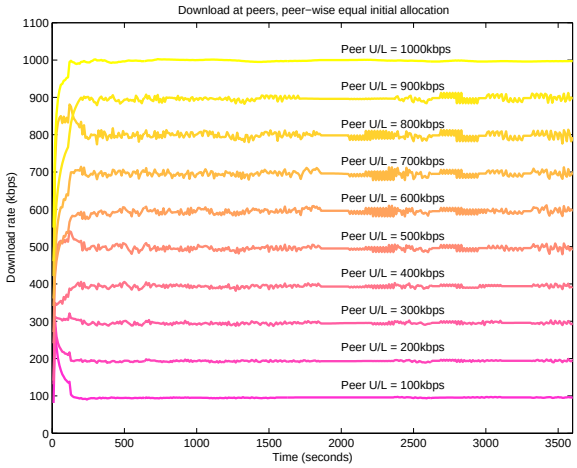


Fig. 6. 10 peers, users can download at a rate that is equal to their upload rate.

D. Fairness of the network

In the previous subsection we argued 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. In this subsection we show that in the saturated regime a strong *pair-wise fairness* holds, *i.e.,* the average amount of bandwidth a pair of users share is equal.

Corollary 1 *If $\gamma_i = 1$ for all i then the allocation scheme* (3) *guarantees pair-wise fairness so that*

$$\overline{\mu_{ji}} = \overline{\mu_{ij}}, \qquad \forall i, j. \tag{16}$$

Proof: Inequality (13) reduces to

$$\overline{\mu_{ij}} = \frac{\mu_i \overline{\mu_{ji}}}{\overline{\mu_i}}. \tag{17}$$

in the considered regime. Since (17) holds for all i, j it follows that $\overline{\mu_{ji}} = \overline{\mu_{ij}}$. ∎

A similar result is given in [19], 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 peer-wise fairness property of (16) 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.

V. Simulations and Experiments

We have implemented a discrete time simulator of the p2p system described in Section III for the purposes of confirming the claims made in this paper. Our simulator permits nodes to initially allocate any feasible amount of upload bandwidth to their peers, although our specific experiments assumed small, equal, non-zero initial contributions between peers. In our experiments, each peer reallocated its upload bandwidth once per second, and our graphs were smoothed over a running average of 10 seconds.

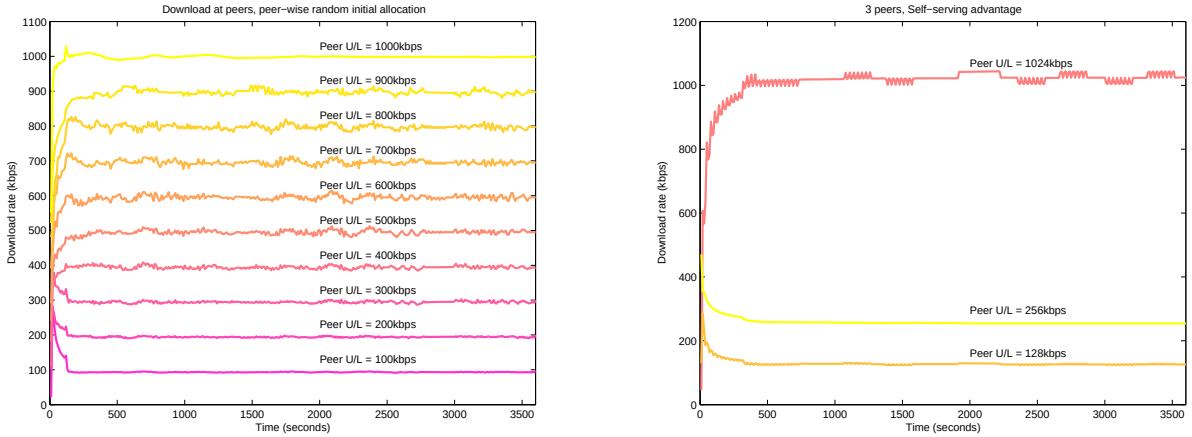
In Section V-A we demonstrate the fairness and incentive results of Section IV. More precisely, we show that the system allocating bandwidth using Equation (3):

- is fair to all peers even in a heterogeneous network environment;
- benefits each cooperating peer by providing instantaneous bandwidth beyond what is a available in a single-user model; and,
- provides benefit and incentive even if some peers allocate bandwidth contrary to (3) (*e.g.,* maliciously).

In Section V-B we demonstrate the effectiveness and implementational efficiency of random linear coding, which securely distributes the information in our system.

A. Fairness and benefit

In Figure 6 the download rate in kbps available to the user of a peer is plotted against the time in seconds. Initially no one has downloaded any content from the other peers and the peers split their bandwidth equally among all other requesting peers. All users corresponding to all peers in the network demand bandwidth starting at time = 0, although their corresponding peers



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

(b) A three peer network. The system is fair even in the case when one peer's upload bandwidth dominates all other upload bandwidths, as is the case here.

Fig. 7. Convergence of the proposed approach, performance in the presence of a dominating peer.

are serving at different rates ($100kbps - 1000kbps$). The distributed system quickly converges to a fair allocation of users receiving what the (corresponding) peers contribute.

Figure 7(a) is setup similar to Figure 6 but in this case the initial bandwidth allocation to other peers is randomly chosen instead of being equally split among them. Again the system converges to a fair allocation of the cumulative service being offered by the network.

In our first experiment, ten users request large files from the system at the same time, but their corresponding peers have different upload capacities ranging from 100kbps to 1000kbps. The download rate provided by the system is plotted in Figure 7(a) as a function of time. We can see that system initially goes through a period where bandwidth allocation among peers looks random, but that it quickly converges to rates that are commensurate with the upload capacities of the various peers. Figure 7(b) shows that this fairness effect exists even in the absence of the non-dominant condition of [18], *i.e.,* even when one peer contributes more than the sum of all other contributions.

Our next few graphs demonstrate the benefits of the proposed system to all collaborating peers commensurate with Theorem 1. For these graphs, we simulated a three peer network comprising 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.

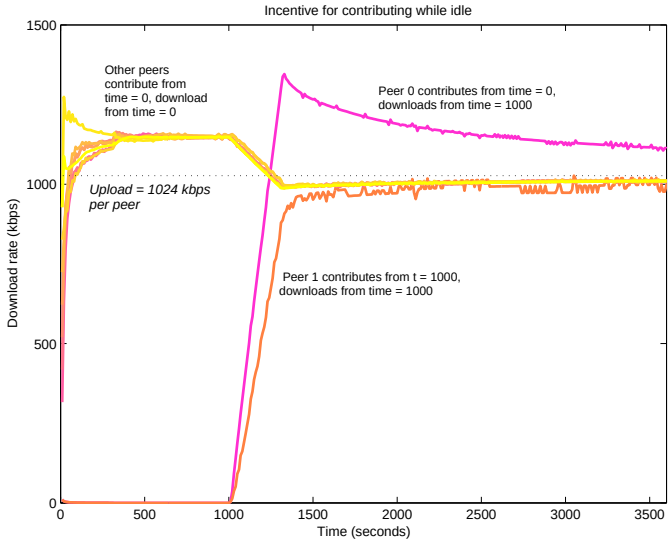
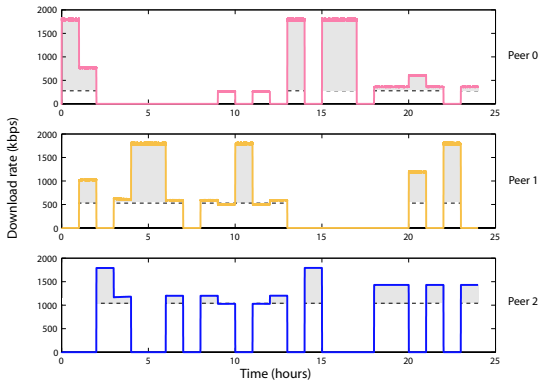
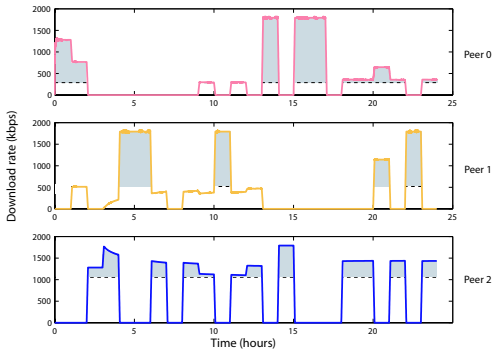


Fig. 8. A peer that contributes while not using the network's resources gets rewarded later. A peer that joins later without contributing earlier suffers comparatively lower download rates.



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



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

Fig. 9. 3 peer network, $\mu_0 = 256kbps$, $\mu_1 = 512kbps$, $\mu_2 = 1024kbps$.

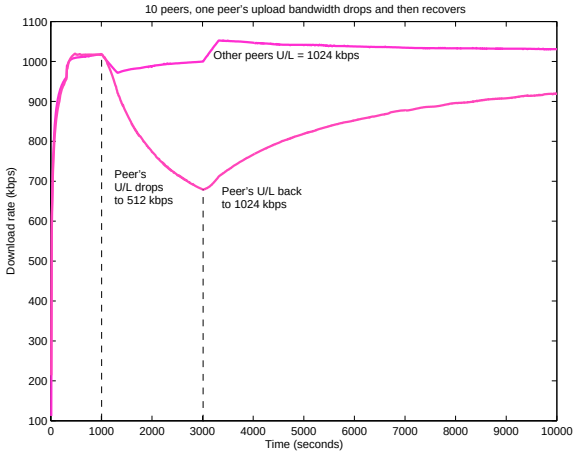


Fig. 10. Simulation scenarios showing incentive to contribute while idle, and the dynamics of the proposed approach.

Figure 9(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 all through 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).

Figure 9(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 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 peer 1 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.

Our next experiment demonstrates that there is incentive for peers to contribute even when their associated users are not downloading from the network, as illustrated in Figure 8. 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 so 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.

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 corresponding to all peers request service throughout. At time $t = 1000$ seconds one

$q \downarrow, m \rightarrow$	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}
GF(2^4)	256	128	64	32	16	8
GF(2^8)	128	64	32	16	8	4
GF(2^{16})	64	32	16	8	4	2
GF(2^{32})	32	16	8	4	2	1

TABLE I

NUMBER OF MESSAGES k REQUIRED (GENERATED) FOR DECODING (ENCODING) 1MB DATA; $k \times k$ COEFFICIENTS FROM $q = GF(2^p)$ COMPRISE THE β COEFFICIENT MATRIX.

$q \downarrow, m \rightarrow$	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}
GF(2^4)	117.28	58.8	30.05	14.99	7.57	3.9
GF(2^8)	34.78	17.52	8.85	4.46	2.29	1.18
GF(2^{16})	10.97	5.53	2.81	1.42	0.72	0.4
GF(2^{32})	3.9	1.96	1	0.51	0.26	0.15

TABLE II

DECODING (ENCODING) TIMES IN SECONDS.

peer's upload bandwidth contribution drops to 512kbps, and there is a consequent decrease in it's download bandwidth as shown in Figure 10. 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 peer will speed up the dynamics of the system; we leave the analysis of this decay as a direction of future work.

B. Performance of random linear codes

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 (2). 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. We believe that this study will also aid in making design decisions in related applications such as network coding based multicast.

In order to reduce any programming inefficiencies we chose to use a Victor Shoup's number theory library [36] in conjunction with the GNU multi-precision library (GMP) [37]. These libraries provide fast software implementations of finite field arithmetic as well as optimized routines for matrix inversion. While the exact decoding times obtained from these experiments will vary with the type of machine and underlying implementations, we believe that these simple experiments put random linear coding in perspective (as compared to uncoded data dissemination). Our experiments were performed on a Pentium 4 dual processor workstation running the Linux operating system, although the code was not parallelized to utilize both the available processors. The encoding and decoding operations are essentially the same, the latter using the inverse of the coefficient matrix in Equation (2), so we only present decoding times in our results.

Table I simply states the number of messages k required to encode 1MB worth of data for different field sizes and message lengths. Thus, for example, if we used finite field $q = GF(2^{32})$ (*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 (2) 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 II it is apparent that lower values of k yield faster decoding. 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. If we give one second for decoding 1 MB of data, then Table II shows that our example of $q = GF(2^{32})$ 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. CONCLUSIONS

In this paper we have proposed a new peer-to-peer application that enables users to overcome slow upload bandwidth bottlenecks when remotely accessing data on their home computers. 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.

We have also shown that our proposed system 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.

To address issues of security and distributed access, our system encrypts and 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. Our adaptation of random linear coding further guarantees authentication of each received packet on the fly and that the number of packets needed for decoding is equivalent to the size of the unencoded data.

A. Future work

It would be especially insightful to implement the proposed system in a dynamic real-time environment, which would have to address tradeoffs between fairness and quick adaptation to changes in the networking environment. Such a system would also require an efficient means of handling rapid changes and modifications of data (in the current incarnation, modifications have to be re-encoded and re-transmitted to the network). Another issue for such a system would be minimizing the amount of meta-data that the user needs to carry around.

We believe that the system we have discussed here will find uses in asymmetric bandwidth situations such as dialup connections, cable modems, ADSL, and wireless cellular internet technologies like GPRS and UMTS. The notions of fairness and incentives developed in the paper are much wider in scope and could find uses in game theoretic settings, markets and economics.

VII. Acknowledgments

The authors wish to thank David Starobinski for editorial inputs. This work was supported in part by NSF CAREER Program through grant ANI/CNS-0238397.

References

[1] Foldershare.com, Bytetaxi, Inc.
[2] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system", *in Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, 2000.* International Computer Science Institute.
[3] M. J. Freedman, E. Freudenthal, and D. Mazires, "Democratizing content publication with Coral", *in Proc. 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04)*, March 2004.
[4] B. Cohen, "BitTorrent", *http://bitconjurer.org/BitTorrent*.
[5] Sherman Networks, "Kazaa", *http://www.kazaa.com*.
[6] Gnutella, *http://gnutella.com*.
[7] Carnegie Mellon University, "Coda file system ",*http://www.coda.cs.cmu.edu/*.
[8] Xerox Parc, "Bayou Project", *http://www2.parc.com/csl/projects/bayou/*.
[9] 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*.
[10] Joanna Glasner, "P2P fuels global bandwidth binge", *http://www.wired.com /news/business/0,1367,67202,00.html*, *Wired*, April 14, 2005.
[11] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment", *IEEE Journal on Selected Areas in Communications*, 2003. *http://www.cs.berkeley.edu/ ravenben/publications/CSD-01-1141.pdf*.
[12] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", *in Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany*, pages 329-350, November 2001.
[13] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility", *in Proc. 18th ACM SOSP'01*, October 2001.
[14] J. Byers, M. Luby, M: Mitzenmacher and A. Rege, "A Digital Fountain approach to reliable distribution of bulk data", *in Proc. of ACM Sigcomm*, September 1998.
[15] D. Qiu, and R. Srikant, "Modelling and performance analysis of bittorrent-like peer-to-peer networks", *in Proc. of ACM Sigcomm*, August 2004.
[16] 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.
[17] 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*.
[18] X. Yang and G. de Veciana, "Service capacity of peer-to-peer networks", *in Proc. of IEEE Infocom*, 2004.
[19] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: service capacity and role of resource rharing policies", *(to appear) in Performance Evaluation - Special Issue on Performance Modeling and Evaluation of P2P Computing Systems*, 2005.
[20] 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.
[21] 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.
[22] 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.
[23] 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).

[24] Landon Cox and Brian Noble. “Samsara: honor among Thieves in peer-to-peer Storage”, *in Proc. of the ACM Symposium on Operating Systems Principles*, October 2003.
[25] L. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan “Chord: A Sclable peer-to-peer lookup service for internet applications”, *in Proc. ACM Sigcomm*, August 2001.
[26] J. Wylie, M. Bigrigg, J. Strunk, G. Ganger, H. Kiliccote and P. Khosla,“Survivable information storage systems”*IEEE Computer*, August 2000.
[27] 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.
[28] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding”, *In Proc. Allerton Conference on Communication, Control, and Computing*, October 2003.
[29] A. Shamir, “How to share a secret”, *Communications of the ACM*, pp. 612-613, 22, 1979.
[30] R. J. Anderson, “The Eternity Service”, *in Proc. of Pragocrypt* , 1996.
[31] M. Waldman, A.D. Rubin, and L.F. Cranor, “Publius: A robust, tamper-evident, censorship-resistant and source-anonymous web publishing system”, *in Ninth USENIX Security Symposium*, August 2000.
[32] Eric W. Weisstein. “Coupon collector’s Problem”, *from MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/CouponCollectorsProblem.html*.
[33] 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.
[34] 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.
[35] T. Migler, K. E. Morrison, M. Ogle, “Weight and rank of matrices over finite fields”, *http://www.calpoly.edu/ kmorriso/Research/weight.pdf*.
[36] V. Shoup,“Number theory library”, *http://shoup.net/ntl*.
[37] “GNU multi precision libary”, *http://www.swox.com/gmp/index.orig.html*.
[38] 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.
[39] 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.
[40] 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.
[41] 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.
[42] S.Agarwal, M. Laifenfeld, A. Trachtenberg, and M. Alanyali, “Using bandwidth sharing to fairly overcome channel asymmetry”, *Information theory and applications workshop, UCSD*, 2006.