Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 968328, 11 pages
http://dx.doi.org/10.1155/2014/968328

Research Article

ISP-Friendly Data Scheduling by Advanced Locality-Aware Network Coding for P2P Distribution Cloud

Yanjun Li,[1] Guoqing Zhang,[1] and Guoqiang Zhang[2]

[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2]*School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China*

Correspondence should be addressed to Guoqiang Zhang; guoqiang@ict.ac.cn

Received 24 October 2014; Accepted 19 November 2014; Published 15 December 2014

Academic Editor: Yoshinori Hayafuji

Peer-to-peer (P2P) file distribution imposes increasingly heavy traffic burden on the Internet service providers (ISPs). The vast volume of traffic pushes up ISPs' costs in routing and investment and degrades their networks performance. Building ISP-friendly P2P is therefore of critical importance for ISPs and P2P services. So far most efforts in this area focused on improving the locality-awareness of P2P applications, for example, to construct overlay networks with better knowledge of the underlying network topology. There is, however, growing recognition that data scheduling algorithms also play an effective role in P2P traffic reduction. In this paper, we introduce the advanced locality-aware network coding (ALANC) for P2P file distribution. This data scheduling algorithm completely avoids the transmission of linearly dependent data blocks, which is a notable problem of previous network coding algorithms. Our simulation results show that, in comparison to other algorithms, ALANC not only significantly reduces interdomain P2P traffic, but also remarkably improves both the application-level performance (for P2P services) and the network-level performance (for ISP networks). For example, ALANC is 30% faster in distributing data blocks and it reduces the average traffic load on the underlying links by 40%. We show that ALANC holds the above gains when the tit-for-tat incentive mechanism is introduced or the overlay topology changes dynamically.

1. Introduction

Peer-to-peer (P2P) file distribution clouds are becoming more popular in recent years. Their attractiveness for content providers is obvious, particularly because of the improved application-level performance and reduced distribution cost. There is, however, a growing recognition that P2P applications are in general "unfriendly" to Internet service providers (ISPs). This is because P2P applications generate enormous traffic [1]. Such rapid growth in P2P traffic raises ISPs' costs in many ways. Firstly, small ISPs have to pay millions of dollars to their provider ISPs for the huge amount of cross domain P2P traffic. Secondly ISPs are forced to frequently upgrade their network infrastructures to cope with ever faster increase in traffic demand. Other costs to ISPs include increasing energy consumption and growing size of P2P cache. From the ISPs' point of view, P2P is an unfair way for content providers to shift their own distribution costs to ISPs.

P2P cloud systems are largely network-oblivious. They operate on overlay networks built on top of underlying physical networks, using little or limited knowledge of the network topology and locality information. To reduce cross domain P2P traffic or P2P traffic in general, we need to improve the efficiency of network resource usage. For example, a large amount of long-distance traffic that imposes heavy stress on the underlying network infrastructure should be avoided.

One way to achieve better use of network resources is to achieve the so-called locality-awareness in the construction of overlay networks as well as in the download process. This has attracted extensive research interests in recent years. P2P applications can now obtain locality knowledge by the reverse-engineering [2–8] and ISP services [9–12], such as P4P. With more and more accurate locality information becoming available, the performance gain arising from the locality-awareness approach will reach its limit.

Another complimentary approach lies in the data scheduling algorithm, which defines how a P2P application propagates data blocks on its overlay network. Traditional

P2P applications use either the random scheduling or the local-rarest-first scheduling. However, they both suffer from the problem of biased distribution of data blocks and consequently limit the utility of locality information. A more recent data scheduling algorithm is the network coding [13]. It simplifies the scheduling process and improves the application-level performance of P2P services [14, 15]. It has been shown that network coding meets multimedia applications [16] and the recently proposed new information-centric networking architecture [17].

We recently introduced the locality-aware network coding (LANC) [18], which can reduce cross domain P2P traffic by as much as 50%. It is because network coding is able to obtain a more balanced distribution of coded data blocks in a P2P system. This increases the chance for a peer to find useful blocks within its neighbourhood. Aided by proper locality knowledge, the probability for a peer to retrieve useful blocks from its proximate neighbors will increase as well. Network coding and LANC, however, suffer from the linearly dependent data blocks problem. In LANC, the linearly dependent data blocks can account for over 10% of all data block transmissions and should be avoided.

In this paper, we propose the advanced locality-aware network coding (ALANC), which improves over our previous work in two facets.

(1) ALANC completely avoids the transmission of linearly dependent blocks that both NC and LANC suffer from.

(2) Aside from the benefit of interdomain P2P traffic reduction, network coding-based scheduling supported by locality information is also capable of alleviating traffic burden on intradomain P2P traffic and thus is effective in reducing P2P traffic in general.

We introduce a simulator to evaluate how ALANC improves P2P and network performance. Our results show that ALANC substantially improves both the application-level performance (good for content providers and end users) and the network-level performance (ISP-friendly). We also demonstrate that ALANC holds the advantages over other scheduling algorithms when an incentive mechanism is introduced or when the overlay network is dynamic.

2. Background

The massive amount of traffic generated by P2P systems raises criticisms from ISPs. Today, relieving P2P traffic burden is a hot topic in the research community, as evidenced by the establishment of the ALTO (Application-Layer Traffic Optimization) Working Group in IETF [19]. Currently there are three approaches to achieve P2P traffic localisation: P2P cache, locality knowledge provision, and data scheduling.

2.1. P2P Cache. The ISPs can use their widely deployed caches [2, 20, 21] to cache P2P traffic so that duplicated data transmissions on backbone networks can be reduced. Cache replacement algorithms, for example, partial caching [20], have been developed to address the characteristics of P2P traffic, which is distinctively different from those of Web traffic, for example, the difference in popularity distribution of objects.

P2P cache is limited by the fact that it is not scalable, because it has to speak various P2P protocols, most of which are proprietary. ISPs in general are not in favour of this solution as it effectively shifts the cost of data distribution from content providers to ISPs themselves [2]. Caching content may also raise legal issues.

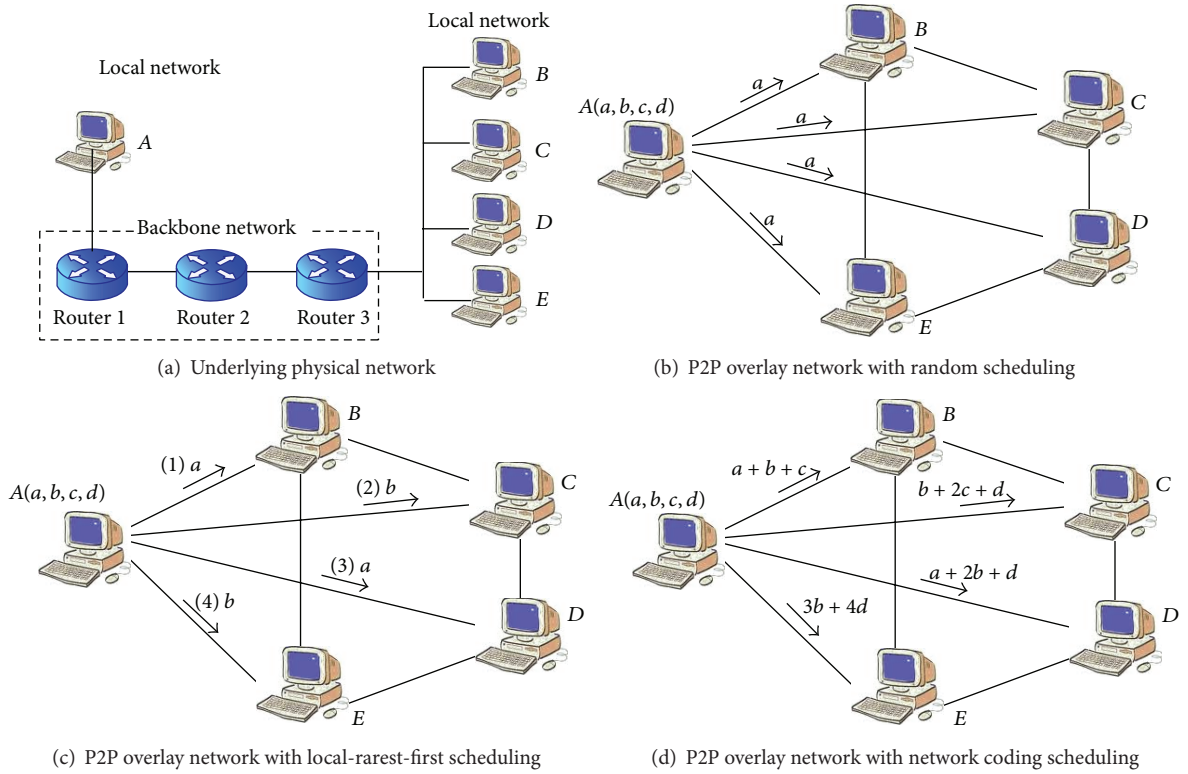
2.2. Locality-Awareness. A more attractive solution is to construct overlay networks based on the locality information of underlying networks. The key of this solution is to acquire accurate knowledge of the locality information of underlying networks.

(1) Reverse-Engineering Techniques. They include active probing, for example, landmark-based proximity identification [6, 7] and network coordinate systems [22–24], and passive inference, for example, identifying a host's autonomous systems number (ASN) by its IP address [2, 4, 25]. Such techniques are inherently limited by the granularity and accuracy of their data sources.

(2) ISPs' Locality Services. ISPs are at the right position to offer the most accurate locality knowledge as services. They are willing to do so because these services allow ISPs and P2P applications to jointly optimize their respective performances and ultimately create a win-win situation between them. A number of such ISP services have been proposed in recent years. For example, the Oracle service [9] provides a peer ranking service based on topological metrics. The P4P [10] proposes an architecture for an ISP to opaquely expose the network distance information without sacrificing its privacy. Such information can then be used by P2P applications to shape their connectivity on the overlay network and choose network-efficient communication patterns. With ISPs participation, the accuracy of locality knowledge has been significantly improved.

2.3. Data Scheduling. In this subsection, we introduce commonly used data scheduling algorithms, along with a motivation example that shows how the utility of locality information could be inherently limited by conventional data scheduling algorithms, whereas this limitation can be to large extent overcome by network coding scheduling.

P2P file distribution applications, such as BitTorrent [26], use data scheduling algorithms to organise the data download process. Figure 1 illustrates an example. Figure 1(a) shows an underlying physical network, where hosts B, C, D, and E are in a local network and they are connected to host A via a series of routers on a backbone network. Figure 1(b) shows a P2P overlay network constructed over the underlying network, where the hosts are registered as application-level peers. The general scenario is that a peer, say A, functions as a server. It holds a data file and aims to distribute the file to other peers on the system. It is highly undesirable, if possible at all, for the server alone to serve all the peers. Instead, the server splits the file into four data blocks a, b, c, and d, and sends data blocks



(a) Underlying physical network

(b) P2P overlay network with random scheduling

(c) P2P overlay network with local-rarest-first scheduling

(d) P2P overlay network with network coding scheduling

Figure 1: An example of P2P content distribution system and data scheduling algorithms.

to peers on demand. This allows the peers to exchange data blocks among themselves and therefore alleviate the stress on the server. Each peer knows a small set of other peers, which form its *neighbourhood*, and it only exchanges data with its neighbours. A peer relies on a data scheduling algorithm to request *innovative data blocks* that the peer does not already have. Existing data scheduling algorithms are as follows.

(1) Random Scheduling. A peer requests a random data block from all innovative blocks within its neighborhood.

As shown in Figure 1(b), the worst case is when the peers B, C, D, and E all request the same data block, say a, from server A. The data block will have to pass through the backbone routers four times. In the optimal case, each of the peers requests a different data block, such that only one copy of the original file passes through the backbone routers. The peers can then exchange data blocks among themselves via local downloading. However the probability for the ideal case is very low, only $4!/4^4 = 9.375\%$. This means for most cases at least one data block has to be transmitted through the backbone network links for more than once.

(2) Local-Rarest-First (LRF) Scheduling. A peer requests the rarest data block among all innovative blocks in its neighbourhood. It is reported [4] that comparing with the random scheduling the LRF scheduling significantly reduces interdomain P2P traffic redundancy.

For the LRF scheduling, if there are multiple neighbours who can offer the same rarest data block, a peer randomly chooses a neighbor. If in this case a peer applies the locality-aware downloading (LAD), that is, it chooses the closest neighbour which is most proximate on the underlying network, it is called the LRF+LAD scheduling.

As shown in Figure 1(c), suppose peer B first requests a data block a from A. Since there are two copies of a in peer C's neighbourhood (A, B, and D), C then requests a rarer data block, say b, from A. Then D determines that in its neighborhood A holds the locally rarest data blocks a, c, and d. So it requests a from A with probability 1/3. Similarly, E requests data block b from A with probability 1/3. Although each of the four peers has requested a locally rarest data block from the server, there is a high chance that they request for the same blocks, in this example blocks a and b. Now, even LRF+LAD is used, the local links between B, C, D, and E can only be used for another round of data block exchanges. After that, the peers will have to make further requests to the server for other data blocks that they do not collectively have. It is clear there is much room for improvement.

(3) Network Coding (NC) Scheduling. Network coding was first proposed as a technology to realize the upper bound of the theoretical multicast capacity predicted by the max-flow min-cut theorem [13, 27, 28]. It is a paradigm shift from the conventional information transmission and processing mode by allowing intermediate nodes to perform arbitrary coding functions on the input data. Network coding has become an active research area [27, 29–33]. Recently there are studies on using the network coding as a data scheduling algorithm for P2P file distribution systems [14–16, 18, 34].

When network coding is used, peers do not transmit the original data blocks. Instead they generate and exchange coded data blocks. Suppose the original file for distribution is split by the server into n data blocks, $X = (x_1, x_2, \ldots, x_n)$. A coded data block is in the form of

$$b = g_b X^T, \tag{1}$$

where g_b is an n-tuple, $(g_1, g_2, \ldots, g_n)$ and is called the global coding coefficient of the coded data block. For a peer i with I coded data blocks, its global encoding coefficients at time t are represented as an $I \times n$ matrix $A_i(t)$. A peer exchanges the global coding coefficients with its neighbours such that two neighbouring peers know each other's global coding coefficients.

When a peer requests a coded data block, it first enumerates its neighbours and constructs a candidate list of neighbouring peers which have innovative data blocks. It then randomly chooses a candidate to make the request.

When a peer receives a request, it generates a new coded data block as follows. Firstly it independently chooses m coded data blocks $(b_1, b_2, \ldots, b_m)$ from its available coded blocks. Secondly it generates m random *local* encoding coefficients $(c_1, c_2, \ldots, c_m)$ and produces a new coded data block as $b = (c_1, c_2, \ldots, c_m)(b_1, b_2, \ldots, b_m)^T$. Thirdly it calculates the global encoding coefficient of the new coded block as

$$g_b = \begin{pmatrix} c_1 & c_2 & \cdots & c_m \end{pmatrix} \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{pmatrix}, \tag{2}$$

where $(g_{i1}, g_{i2}, \ldots, g_{in})$ is the global coding coefficient of the coded block b_i. Then it sends the new coded block and the global coding coefficient to the requestor. The parameter m is called the encoding density. It directly relates to the encoding complexity. When $m = 1$, it is equivalent to not using the network coding. In case a peer has less than m coded blocks, it uses all available blocks.

Finally, when a peer receives n linearly independent coded blocks, it can decode the original data blocks as $X = A^{-1}Y$, where Y is the vector of the n coded blocks, and A^{-1} is the inversion of the matrix induced from the global encoding coefficients of the n coded blocks.

Recently it is shown [14, 34] that network coding can be used as a data scheduling algorithm to improve the application-level performance of P2P file distribution applications because of its simplified data scheduling process. In network coding, a requestor needs only to choose a neighbouring peer to send its request, but it does not need to determine which data blocks of the peer to request. In contrast, the LRF algorithm requires a peer to analyze the frequency distribution of all data blocks within its neighbourhood whenever it makes a request. However, there are still insufficient incentives for the wide deployment of network coding-based P2P systems, largely because of the concern about its computation overhead.

Only recently, network coding's potential for efficient resource utilization is gradually being recognized. It was shown [14] that NC performs well in overlay network topologies with bad cuts, capable of reducing traffic between clusters. However, this study focused on the application-level performance, not providing any quantitative evaluation of the network-level performance, such as resource utilization efficiency. Recently, we began to recognize that network coding can be an effective way to reduce cross domain P2P traffic [18]. In [35], the authors also developed similar idea of using network coding to reduce congestion in networks in parallel with our work. Their paper, however, is for the P2P streaming scenario. P2P streaming has strict requirement on the delivering rate to each receiver, which is different from P2P file distribution.

Figure 1(d) exemplifies that network coding can significantly improve the utility of locality information and hence achieve more efficient network resource usage than other data scheduling algorithms. With network coding, the server A responds to each data request from the peers with a distinct coded data block. It is known [30] that with the finite field as large as $F(2^8)$ or $F(2^{16})$, there is a high probability that the four coded blocks sent to the four peers are linearly independent. Hence there is no need for the peers to make further request to the server through backbone links. They only need to exchange the coded data blocks among themselves in the local network by the support of locality-aware downloading and then use the coded blocks to reconstruct the original file. This implies that network coding can achieve highly efficient use of the underlying network resources.

(4) Summary of Data Scheduling Algorithms. To summarize, the above example exemplifies that data scheduling algorithms can have significant impact on P2P traffic burden. With traditional data scheduling such as random and LRF, the probability that the same block travels the same backbone links multiple times is not negligible. But if data is scheduled by network coding, this probability could be much lower if locality knowledge is also used in the download decision process.

3. Our Network Coding-Based Algorithms

3.1. Locality-Aware Network Coding (LANC). Recently we introduce the locality-aware network coding (LANC) [18] which incorporates the locality-aware downloading policy in the data scheduling process.

In the original network coding, when a peer makes a request, it first constructs a list of neighbours who have innovative coded blocks. It then randomly chooses a neighbour to make the request. In our LANC, the peer uses the locality-aware downloading policy to select from its candidate list a neighbouring peer which is most proximate to itself on the underlying physical network. We showed [18] that this improvement, although simple, is remarkably more advanced than all existing scheduling algorithms in reducing the interdomain traffic redundancy for P2P file distribution applications.

3.2. Problem of Linearly Dependent Coded Data Blocks. We observed that for the LANC over 10% of all transmitted coded

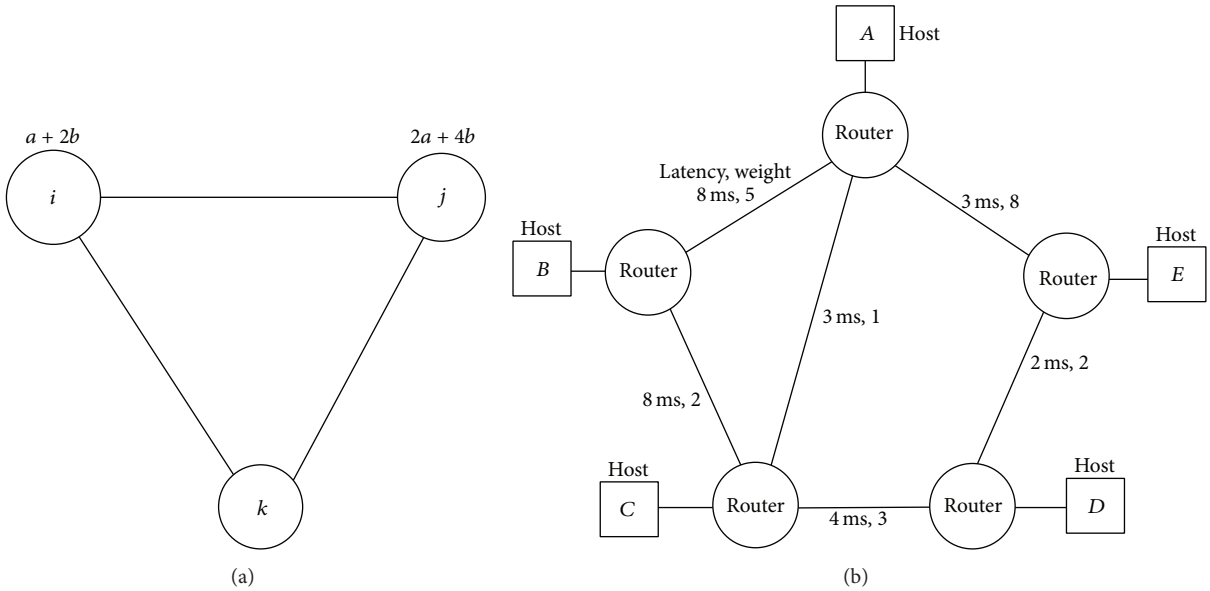


Figure 2: Two examples. (a) Occurrence of linear dependent data blocks. (b) An underlying network.

data blocks are linearly dependent. These data blocks are not useful for reconstructing the original data file and should be avoided.

Figure 2(a) shows an example. Suppose three peers i, j, and k form a triangular overlay topology. Peer i initially has a coded data block $(a + 2b)$, and it sends j a newly coded block $(2a + 4b)$. Then k determines both i and j have an innovative block, so it sends requests to the two peers simultaneously. This will lead to linearly dependent blocks transmitted to peer k.

In general, a peer k will receive linearly dependent blocks if a number of peers $i, i + 1, \ldots, j$ all have innovative blocks for k and the number of requests, r, that k sends to the peers at time t satisfied the following:

$$r > \begin{vmatrix} A_i(t) \\ \vdots \\ A_j(t) \\ A_k(t) \end{vmatrix} - \left|A_k(t)\right|, \tag{3}$$

where $||$ denotes the rank of a matrix.

It is reported [36] that small-world network can cause nonnegligible linearly dependent blocks, such as those constructed with locality information. This problem becomes more serious when locality-aware downloading is used. This is because a few innovative blocks circle around a peer's neighbors, making the number of neighbors who can offer innovative blocks to the peer far exceed the actual number of innovative blocks these neighbors can provide to the peer.

3.3. Advanced Locality-Aware Network Coding (ALANC). In this paper, we introduce the advanced locality-aware network coding (ALANC). It not only incorporates the locality-aware downloading policy, but more importantly completely avoids the problem of linearly dependent coded data blocks.

In the original network coding, a peer responds to a request by transmitting a new coded data block and its global encoding coefficient at the same time. In ALANC, a peer decouples the transmission of the global encoding coefficient and the coded data block. It first transmits the global encoding coefficient of the new coded data block. The peer sends the coded block only after it receives a message from the requestor which confirms the linear independency of the coded block.

This simple approach, however, only ensures that the sender with innovative blocks does not accidentally generate blocks that are linearly dependent with the already available blocks in the receiver. There is a coordination problem. Suppose a peer generates a number of requests to its neighbours. When it receives a global encoding coefficient from a neighbour, it may determine that the coefficient is linearly independent of its own global encoding coefficients and then replies to the neighbour with a confirmation message. But it is possible that when it receives the coded block from the neighbour, the block is not linearly independent anymore because other blocks have just arrived. To solve this problem, we propose that a requestor, say peer i, maintains not only the global encoding coefficients $A_i(t)$ of its available blocks, but also the global encoding coefficients $T_i(t)$ of the expected blocks of which confirmation messages have been sent. The requestor i can then determine that a received global encoding coefficient g_b is linearly independent of both its available blocks and its expected blocks if

$$\begin{vmatrix} g_b \\ A_i(t) \\ T_i(t) \end{vmatrix} > \begin{vmatrix} A_i(t) \\ T_i(t) \end{vmatrix}. \tag{4}$$

Table 1: Reduction of interdomain P2P traffic.

Scheduling algorithm	Interdomain traffic redundancy	Percentage of linearly dependent blocks
LRF+LAD	12.71	—
NC	14.91	3.75%
LANC	5.87	10.0%
ALANC	5.57	0

Similarly, when constructing the candidate list, i can infer that a neighboring peer j has innovative blocks for itself if

$$\begin{vmatrix} A_j(t) \\ A_i(t) \\ T_i(t) \end{vmatrix} > \begin{vmatrix} A_i(t) \\ T_i(t) \end{vmatrix}. \quad (5)$$

3.4. Reduction of Interdomain P2P Traffic. To evaluate the effect of our network coding-based algorithms on reducing interdomain P2P traffic, we run the simulation as detailed in our earlier work [18]. Table 1 shows the results. The first measure is the interdomain traffic redundancy, which is defined as the ratio of the actual number of interdomain blocks to the theoretical optimal number of interdomain blocks that are required for the distribution of a data file to peers located in different domains. The optimal situation is when only one copy of the original file is transmitted to each domain. In that case the interdomain traffic redundancy is 1. When compared with LRF+LAD, our network coding-based scheduling algorithms LANC and ALANC reduce interdomain P2P traffic by over 50%. This is a remarkable achievement considering the shear volume of traffic generated by P2P file distribution applications. On the other hand, network coding alone cannot reduce the interdomain P2P traffic. It only offers the potential for P2P traffic reduction. To realize the potential, locality-aware downloading is necessary.

The second measure is the percentage of linearly dependent blocks. Although locality-aware downloading is necessary to realize P2P traffic reduction, it aggravates the linearly dependent data block problem. The percentage of linearly dependent data blocks increases from 3.75% in NC to 10% in LANC, which is a huge waste of network resources. As expected, ALANC completely avoids the problem of linearly dependent blocks.

4. Performance Evaluation

Here we introduce our simulator and present our simulation results. We show that, in addition to interdomain traffic reduction, ALANC also significantly improves the application-level as well as the network-level performance of P2P file distribution.

4.1. Our Simulator. Our simulator constructs a P2P file distribution system on a real ISP's router network. The ISP is Exodus Communications in USA and the network's autonomous system number is AS3967. Data of the router network is provided by the Rocketfuel project [37]. The data file 3967.r1.cch is used. The network has 353 routers and 820 links between the routers. The average shortest path length between a pair of routers is 5.7 hops and the maximum degree, or number of links, of a router is 17. According to the Rocketfuel data, the network's routers are classified as backbone routers and access routers.

Table 2: Selected neighbours of peer A in Figure 2(b).

Routing protocol	Proximity measure	Selected neighbours
OSPF	Latency	B, C, E
OSPF	HOP	C, D, E
RIP	Latency	B, C, D
RIP	HOP	C, D, E

(1) Construction of Overlay Networks. In our simulation we assign 2000 overlay peers to the 262 access routers uniformly. Each peer is then connected to 5 other peers, which, for a probability p, are chosen as the most proximate to the peer on the underlying network and for a probability $1 - p$ are chosen randomly. On the resulted overlay network, each peer on average has $5 \times 2 = 10$ neighbours. We set $p = 0.7$; that is, 70% of connections between peers are based on locality proximity.

In real P2P systems, locality information can be provided by reverse-engineering or ISPs services as discussed in Section 2. In our simulation the proximity between two peers is determined by the following two factors: (1) the routing path between the peers' access routers on the underlay network, which is decided by the intradomain routing protocols, that is, RIP or OSPF, and (2) the proximity measure, which can be the number of hops of the routing path (HOP), or the sum of link latency along the routing path (Latency). Figure 2(b) shows an example of underlay network, where each link's latency and weight values are known. Table 2 shows that the routing protocols and the proximity measures can affect a peer's choice of its overlay neighbours.

We construct four different overlay topologies using combinations of the routing protocols and proximity measures and run simulations on all of them. This allows us to test whether our simulation results are sensitive to overlay topologies.

(2) Estimation of Link Attributes. The Rocketfuel data only provide the latency value of some of the links. We estimate the latency of the other links as in the following. Based on the geographic location of the routers in the Rocketfuel data, we obtain the distance between two access routers using Google's map service, assuming cables are placed along the shortest geodesic path between the routers. Then we divide the distance by the speed that digital signal travels along optic fiber, that is, 2/3 of the light speed in vacuum.

To use the OSPF routing protocol, we need to assign a weight value to each link. There are three categories of links: (a) links between access routers, (b) links between access routers and backbone routers, and (c) links between backbone routers. The Rocketfuel data provide the weight value for links of the third category. We assign a set of random

Table 3: Performance for static overlay networks.

Routing protocol	Proximity measure	Scheduling algorithm	Distribution time	Server load	Router stress	Link stress
OSPF	Latency	Random	482^{+11}_{-14}	270^{+32}_{-17}	2000^{+16}_{-21}	617^{+7}_{-9}
		LRF	492^{+13}_{-13}	257^{+18}_{-14}	2065^{+26}_{-17}	645^{+11}_{-7}
		LRF+LAD	409^{+32}_{-22}	250^{+35}_{-41}	1769^{+39}_{-34}	518^{+17}_{-15}
		NC	433^{+15}_{-18}	206^{+22}_{-30}	1968^{+30}_{-13}	581^{+11}_{-5}
		ALANC	276^{+16}_{-19}	194^{+23}_{-11}	1265^{+15}_{-34}	301^{+6}_{-15}
OSPF	HOP	Random	518^{+26}_{-24}	185^{+10}_{-9}	2058^{+53}_{-40}	642^{+23}_{-17}
		LRF	537^{+36}_{-28}	176^{+7}_{-9}	2155^{+36}_{-40}	684^{+15}_{-17}
		LRF+LAD	439^{+21}_{-20}	178^{+16}_{-12}	1849^{+57}_{-43}	552^{+24}_{-18}
		NC	446^{+17}_{-27}	157^{+14}_{-23}	2033^{+70}_{-38}	605^{+23}_{-12}
		ALANC	300^{+25}_{-36}	144^{+15}_{-9}	1366^{+88}_{-79}	344^{+38}_{-34}
RIP	Latency	Random	489^{+23}_{-17}	195^{+14}_{-9}	1873^{+15}_{-14}	563^{+6}_{-7}
		LRF	485^{+17}_{-35}	197^{+17}_{-19}	1918^{+23}_{-47}	582^{+10}_{-20}
		LRF+LAD	427^{+33}_{-41}	194^{+3}_{-4}	1703^{+47}_{-83}	490^{+20}_{-36}
		NC	407^{+13}_{-6}	153^{+14}_{-17}	1819^{+28}_{-30}	514^{+9}_{-9}
		ALANC	267^{+26}_{-10}	148^{+11}_{-11}	1235^{+34}_{-50}	288^{+57}_{-22}
RIP	HOP	Random	498^{+17}_{-30}	257^{+33}_{-18}	1896^{+43}_{-63}	573^{+18}_{-28}
		LRF	486^{+17}_{-13}	257^{+19}_{-27}	1935^{+31}_{-21}	589^{+14}_{-9}
		LRF+LAD	434^{+12}_{-12}	228^{+53}_{-30}	1731^{+27}_{-35}	501^{+12}_{-15}
		NC	418^{+20}_{-22}	185^{+27}_{-14}	1841^{+61}_{-50}	522^{+23}_{-17}
		ALANC	282^{+19}_{-25}	198^{+15}_{-29}	1262^{+74}_{-83}	300^{+32}_{-36}

weight values to the first category and another set of random values to the second category. It is done under the condition that the ratio of the average link weight among the three categories is 1 : 5 : 15. This ratio is in accordance with the link capacity assumption in [38].

(3) Operation of P2P File Distribution. At the beginning of a simulation, a server holds the original data file which is divided into 100 data blocks. The server is randomly chosen among the 2000 peers. At every time unit of the simulation, each peer attempts to download an innovative block from its neighbouring peers. The simulation stops when no new download attempt can be made and there is no data block in transmission.

The number of blocks a peer can concurrently download and upload is constrained by its download and upload capacity, respectively. For example, when a peer's upload capacity is saturated, the peer can no longer accept new data requests until part of its upload capacity is freed. In our simulation peers can upload and download 3 blocks simultaneously.

When a peer downloads a data block from another peer, the simulator computes the transmission latency between the two peers on the underlying router-level network (along a path determined by the routing protocol in use) and records the arriving time of the data block.

For network coding algorithms, we set the encoding density parameter as ALL, that is a peer uses all of its available blocks to generate a new coded block.

(4) Performance Metrics. During the simulation, we record the total numbers of data blocks passing through each router and each link, the time when a peer finishes downloading all data blocks, the number of peers that are unable to download all data blocks, and the number of blocks served by the server. These data allow us to compute the following performance metrics:

(i) application-level performance metrics:

  (a) distribution time: the average time for a peer to finish its downloads,

  (b) server load: the number of data blocks the server transmits during the file distribution session,

  (c) number of peers unable to finish their downloads due to the incentive mechanism or network dynamics;

(ii) network-level performance metrics:

  (a) router stress: the average number of data blocks a router transits during the file distribution session, including access routers and backbone routers,

  (b) link stress: the average number of data blocks a link transits.

For each overlay topology and each scheduling algorithm, we run the simulation for ten times. All results presented in this paper are the average over ten simulations.

4.2. Performance for Static Overlay Networks. Firstly we run the simulation for static overlay networks, where all peers are preexisting at the beginning of file distribution session, and a peer will remain on the overlay networks after it finishes its own downloading such that it can still exchange data blocks with other peers. Table 3 compares the performance of five scheduling algorithms for each of the four overlay topologies.

We observed that ALANC substantially reduces the link stress. Comparing with the random scheduling, it nearly halves the link stress. In [4], the authors proposed LRF as an effective approach to localize P2P traffic in locality-aware overlay networks. We observed that, even compared with LRF+LAD, a data scheduling that improves over LRF, ALANC can still reduce the link stress by about another 40%. Note that reduced link stress also means reduced traffic in general. ALANC also produces the best performance for other metrics including reduced router stress, reduced distribution time, and reduced server load.

We observe that the locality-aware downloading element has different effects for LRF+LAD and ALANC. For example, LRF+LAD reduces link stress by less than 20% comparing with LRF, whereas ALANC reduces link stress by around 45% comparing with NC. This is because the data blocks in a peer's proximate neighbourhood (i.e., neighbours which are closer to the peer on the underlying network than other neighbours) for LRF are not as diverse as for NC. Therefore there is limited improvement when LRF+LAD chooses proximate neighbours. In contrast ALANC realizes the full potential of NC for traffic localization. Firstly, the random coding of NC substantially increases the diversity of coded data blocks in a peer's proximate neighbourhood. Secondly, the avoidance of linearly dependent blocks ensures that such diversity is truly useful. And thirdly the locality-aware downloading chooses the proximate neighbours.

We observe similar results for all the four overlay topologies which are constructed using different routing protocols and proximity measures. This means ALANC is not sensitive to overlay topology. In the following we only consider one overlay network (OSPF + Latency).

4.3. Performance for Tit-for-Tat Incentive Mechanism. An important issue for P2P applications is that many users are "free-riders," who leech the system without contributing resources to others. In order to discourage this behavior, many P2P applications have introduced incentive mechanisms.

In our simulation we consider the tit-for-tat incentive mechanism introduced by BitTorrent. In addition to the upload and download capacity constraints, a peer i will not accept a request from neighbour j if the number of blocks i has uploaded to j minus the number of blocks i has downloaded from j exceeds a threshold value C. Only after the upload-download imbalance comes under C will i reconsider any request from j.

Figure 3 shows the simulation results for the performance metrics as functions of the threshold value C. We observe that ALANC outperforms other data scheduling algorithms by remarkable margins when the tit-for-tat incentive mechanism is implemented.

Smaller values of C place more stringent constraint on traffic balance between two peers. Figure 3 shows that when C is as small as 4, more than half of the peers cannot finish their downloading for the random, LRF, and LRF+LAD algorithms. This is because when these algorithms are used, a peer is more easily to end up with a situation where none of its neighbours is interested in its available data blocks. In this case the peer cannot download any block because it has nothing to exchange and therefore is wrongly punished by the tit-for-tat mechanism. The longer the file distribution process lasts, the higher the probability for this situation to happen is.

In sharp contrast, NC and ALANC are able to sharply reduce the unfinished peers by two orders of magnitude. Again this is because network coding algorithms can increase the diversity of coded blocks among neighbors. Thus network coding algorithms allow the tit-for-tat mechanism to focus on punishing the real free-riders.

4.4. Performance for Dynamic Overlay Networks. In real P2P systems, the overlay networks are often highly dynamic, where peers join and leave the network with high frequency. In our simulation we consider the following scenarios of dynamic overlay networks.

Initially there is no peer on the overlay network. When a peer joins the network, it is connected to five other peers in the same way as above.

(i) Scenario I: for every 40 time units, 100 peers join the network in a batch. A peer leaves the network after 40 time units since it finishes its downloading. The server is always available in the network.

(ii) Scenario II: it is the same as scenario I, except that the server leaves the network after 40 time units since all the peers join the network.

(iii) Scenario III: it is the same as scenario I, except that the server leaves the network after serving 120 blocks.

(iv) Scenario IV: all peers are present in the network at the beginning. After finishing its downloads, a peer leaves the network with probability q. In this scenario we consider two settings with $q = 0.7$ and $q = 0.5$.

Table 4 shows the simulation results. When calculating the average distribution time, unfinished peers are excluded. When there are >500 unfinished peers, we do not calculate the average distribution time.

We observe that, for dynamic scenarios, ALANC outperforms other data scheduling algorithms in most cases. For scenarios I and II where the server leaves the network at some stage, ALANC enables most peers to finish their downloading, whereas the random and LRF+LAD algorithms heavily depend on the availability of the sever. For scenarios IV where finished peers leave at a given probability, ALANC halves the average link stress. In general, ALANC makes a P2P file distribution system more robust for dynamic overlay networks, in particular for sudden server departure.

5. Conclusion

Reducing P2P traffic burden is a critical challenge for the continuing success of P2P systems. In this paper, we proposed

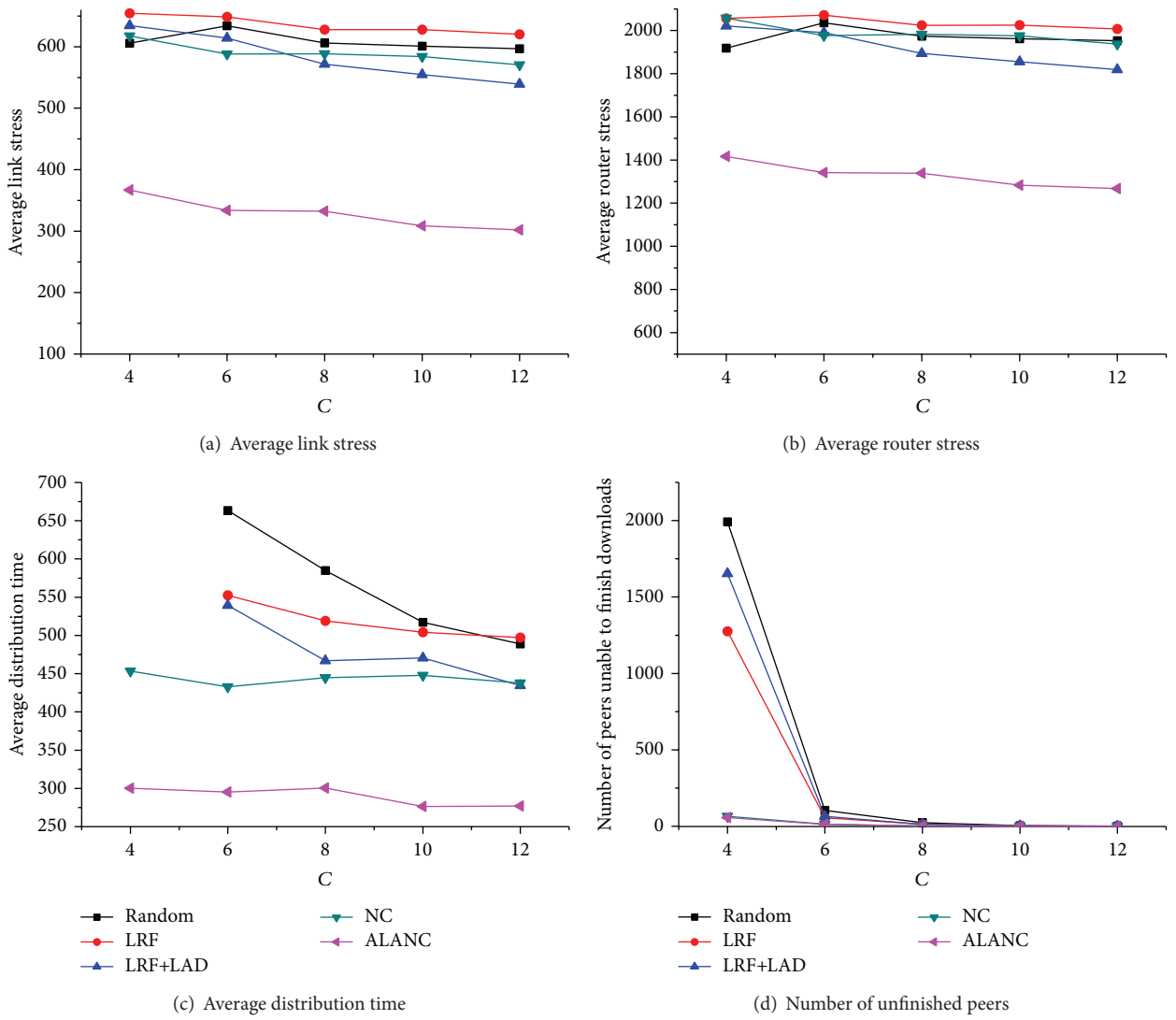


Figure 3: Performance for the tit-for-tat incentive mechanism. C is the threshold value.

ALANC as a promising data scheduling algorithm to alleviate the heavy traffic burden imposed by P2P file distribution applications. There is a limit for conventional data scheduling algorithms to utilize the locality information. Such limitation is largely lifted by ALANC, which is based on network coding, incorporates the locality-aware downloading, and avoids the problems of linearly dependent blocks.

Our results show that, comparing with existing scheduling algorithms, our ALANC can reduce interdomain P2P traffic by 50%, whereas compared with our previously proposed LANC which can incur as much as 10% linearly dependent blocks ALANC completely avoids the problem of linearly dependent blocks. We also introduce a simulator to evaluate the performance benefits of the algorithm. Our results show that ALANC also substantially improves the application-level as well as the network-level performance. Compared with the best approach that does not use network coding, ALANC can reduce the P2P traffic in general by over 40%. And it performs well when an incentive mechanism is used or when overlay networks are highly dynamic. The only cost is the encoding and decoding overhead imposed on end users.

We propose that P2P file distribution system based on ALANC is beneficial for all parties involved. It improves the application-level performance that matters to content providers and end users. More importantly, it improves the utilization of underlying network resources and therefore is friendly to ISPs. Lighter interdomain and intradomain P2P traffic burdens reduce ISPs' operating cost, improve their traffic engineering ability, and relieve their need for frequent and costly network infrastructure upgrades.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Table 4: Performance for dynamic overlay networks.

Scenario	Scheduling algorithm	Distribution time	Unfinished peers	Server load	Router stress	Link stress
I	Random	985	186	194	2244	726
	LRF	984	145	175	2309	754
	LRF+LAD	944	59	155	1999	618
	NC	869	203	173	2402	770
	ALANC	752	193	229	1661	481
II	Random	—	2000	117	2153	704
	LRF	—	545	140	2169	705
	LRF+LAD	—	1614	114	1853	577
	NC	850	214	144	2348	752
	ALANC	791	129	131	1723	503
III	Random	—	2000	120	2228	727
	LRF	—	2000	120	2246	740
	LRF+LAD	—	2000	120	1765	541
	NC	883	206	120	2445	785
	ALANC	762	166	120	1684	488
IV $q = 0.7$	Random	553	28	255	2000	617
	LRF	505	6	298	2055	641
	LRF+LAD	454	17	274	1825	542
	NC	427	2	222	1933	570
	LANC	287	3	198	1262	300
IV $q = 0.5$	Random	473	0	289	1957	599
	LRF	504	0	272	2059	643
	LRF+LAD	434	0	281	1808	535
	NC	435	0	214	1955	576
	LANC	283	0	189	1258	298

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant nos. 61100178, 61174152, and 61303243 and the Key Program of the National Natural Science Funds of China (Grant no. 61331008). Also, this work is funded by Project BK20141454 supported by NSF of Jiangsu Province of China.

References

[1] G. Shen, Y. Wang, Y. Xiong, B. Zhao, and Z. Zhang, "HPTP: relieving the tension between ISPs and P2P," in *Proceedings of the 6th International Workshop on Peer-To-Peer Systems (IPTPS '07)*, Bellevue, Wash, USA, 2007.

[2] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should ISPs fear peer-assisted content distribution?" in *Proceedings of the Internet Measurement Conference (IMC '05)*, 2005.

[3] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D. D. Yao, "Optimal peer selection for P2P downloading and streaming," in *Proceedings of the IEEE INFOCOM*, pp. 1538–1549, March 2005.

[4] R. Bindal, P. Cao, W. Chan et al., "Improving traffic locality in BitTorrent via biased neighbor selection," in *Proceedings of the IEEE Internationa26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)*, p. 66, July 2006.

[5] Y. H. Liu, X. M. Liu, L. Xiao, L. M. Ni, and X. D. Zhang, "Location-aware topology matching in P2P systems," *Proceedings of the 23rd AnnualJoint Conference of the IEEE Computer and Communications (INFOCOM '04)*, vol. 4, pp. 2220–2230, 2004.

[6] X. Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A construction of locality-aware overlay network: mOverlay and its performance," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 18–28, 2004.

[7] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '02)*, vol. 3, pp. 1190–1199, New York, NY, USA, 2002.

[8] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*, pp. 363–374, August 2008.

[9] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P users cooperate for improved performance?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 29–40, 2007.

[10] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4p: provider portal for applications," in *Proceedings of*

the ACM Conference on Data Communication (SIGCOMM '08), pp. 351–362, August 2008.

[11] G. Q. Zhang, M. D. Tang, S. Q. Cheng et al., "P2P traffic optimization," *Science China Information Sciences*, vol. 55, no. 7, pp. 1475–1492, 2012.

[12] N. Magharei, R. Rejaie, I. Rimac, V. Hilt, and M. Hofmann, "ISP-friendly live P2P streaming," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 244–256, 2014.

[13] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[14] C. Gkanisidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of the IEEE INFOCOM*, 2005.

[15] M. Wang and B. Li, "Lava: a reality check of network coding in peer-to-peer live streaming," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1082–1090, May 2007.

[16] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network coding meets multimedia: a review," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1195–1212, 2013.

[17] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design: Architecture, Algorithms, Applications (NoM '12)*, pp. 31–36, June 2012.

[18] G. Zhang and S. Cheng, "LANC: locality-aware network coding for better P2P traffic localization," *Computer Networks*, vol. 55, no. 6, pp. 1242–1256, 2011.

[19] R. Alimi, R. Penno, and Y. Yang, "ALTO Protocol," draft-ietf-alto-protocol-27.txt, 2014, http://datatracker.ietf.org/doc/draft-ietf-alto-protocol.

[20] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1447–1460, 2008.

[21] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak, "Cache replacement policies revisited," in *Proceedings of the 4th GP2P Workshop*, Chicago, Ill, USA, April 2004.

[22] T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proceedings of the IEEE 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 1, pp. 170–179, 2002.

[23] L. Tang and M. Crovella, "Virtual landmarks for the Internet," *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC '03)*, pp. 143–152, 2003.

[24] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network cooridinate system," in *Proceedings of the Annual Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '04)*, Portland, Ore, USA, August 2004.

[25] C. Cramer, K. Kutzner, and T. Fuhrmann, "Bootstrapping locality-aware P2P networks," in *Proceedings of the 12th IEEE International Conference on Networks (ICON '04)*, 2004.

[26] http://www.bittorrent.com.

[27] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[28] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[29] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proceedings of the 15th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 286–294, June 2003.

[30] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '03)*, July 2003.

[31] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proceedings of the Allerton Conference on Communicaiton, Control, and Computing*, 2003.

[32] P. Chou and Y. Wu, "Network coding for the Internet and wireless networks," Tech. Rep. MSR-TR-2007-70, 2007.

[33] Y. Z. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 107–120, 2004.

[34] C. Gkantsidis, J. Miller, and P. Rodriguez, "Anatomy of a P2P content distribution system," in *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS '06)*, Santa Barbara, Calif, USA, 2006.

[35] D.-C. Tomozei and L. Massoulie, "Flow control for cost-efficient peer-to-peer streaming," in *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, San Diego, Calif, USA, March 2010.

[36] T. Small, B. Li, and B. Liang, "Topology affects the efficiency of network coding in peer-to-peer networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 5591–5597, Beijing, China, May 2008.

[37] Rocketfuel Project, http://research.cs.washington.edu/networking/rocketfuel.

[38] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, June 2009.