Evaluation of Network Impact of Content Distribution Mechanisms

Shibsankar Das   Jussi Kangasharju
Department of Computer Science, Darmstadt University of Technology, Germany

Abstract

Efficient large-scale content distribution continues to be an important problem, due to the increasing popularity of multimedia content and wide-spread use of peer-to-peer file sharing. In this paper, we evaluate the impact of different content distribution mechanisms on the network level, based on how much wide area traffic they generate. We consider traditional unicast, content distribution networks (CDN), BitTorrent, and multicast. We develop an analytical model for describing the amount of network traffic generated by the mechanisms and provide experimental results. Our results indicate that BitTorrent can be quite wasteful of network resources, whereas the traditional mechanisms have a significantly lower cost. Based on our results, we propose a series of modifications to BitTorrent which enable us to fully exploit the power of a peer-to-peer content distribution system and result in a network cost often lower than in CDNs and comparable to multicast-based distribution.

1 Introduction

Large-scale content distribution has received a lot of attention in the recent years and it remains an important topic. As multimedia files like audio and video become more popular and software packages become larger and larger, the network has to cope with increased amounts of traffic. Several different content distribution mechanisms have been developed, in order to reduce the network impact of distributing large files.

In this paper, we evaluate several popular content distribution mechanisms in terms of how much wide area network traffic they generate. Based on the results of our evaluation, we propose improvements to the mechanisms in order to improve their performance. The contribution of this paper is therefore two-fold. Our first contribution is the evaluation of network impact of different content distribution mechanisms and our second contribution is a proposal for a new content distribution architecture, which is based on existing mechanisms with improvements based on the lessons learned in our evaluation.

Our focus is on evaluating the cost of a content distribution mechanism, in terms of the amount of wide-area network traffic it generates. Rapidly increasing amounts of network traffic, especially from peer-to-peer file sharing networks, increase the costs for the ISPs that are involved in transporting this traffic. Therefore, it is of high interest to develop efficient content distribution mechanisms which minimize the amount of traffic, while still satisfying all the user demands.

We study five different content distribution mechanisms: unicast from a single server, Content Distribution Network (CDN), multicast, BitTorrent, and our proposed Peer-Assisted Content Distribution Network. Unicast, CDN, and BitTorrent are currently widely used and our evaluation shows their performance relative to each other. Multicast represents the lower bound achievable by any content distribution mechanism. Finally, based on our evaluation, we propose the Peer-Assisted CDN, which is an improvement to BitTorrent, and its performance evaluation highlights the importance of how to exploit peers in a peer-to-peer content distribution system.

This paper is organized as follows. In Section 2 we provide an overview of the different content distribution mechanisms. In Section 3 we present an analytical model for evaluating the network impact and define our cost model. Section 4 describes the setup of our experimental evaluation and Section 5 presents the results of the evaluation. We discuss the results in Section 6. Section 7 presents related work and Section 8 concludes the paper.

2 Content Distribution Mechanisms

In this section, we will present an overview of the different content distribution mechanisms we will study in this paper. Our focus is on delivering stored content, typically large files. We compare five different mechanism in this paper: unicast distribution from a single server, a content distribution network, multicast, BitTorrent, and finally our proposal, a peer-assisted content distribution network. In the following, we will give a short overview of each of these mechanisms.

INFOSCALE '06. Proceedings of the First International Conference on Scalable Information Systems, May 29-June 1 2006, Hong Kong

2.1 Origin Server Only

The simplest form of content distribution is single server distribution, i.e., the file is available only on one server. This is the traditional way how content has been distributed, with FTP and later with Web servers.

2.2 Content Distribution Network

A content distribution network (or CDN), such as Akamai [2], acts on behalf of a content provider and provides a platform for high-demand content delivery. A content distribution network typically consists of two parts: a redirection architecture and a content delivery architecture.

The content delivery architecture consists of a large number of content servers, placed on the edges of the network in locations where they are close to the clients. These content servers contain the content that the content provider wishes to distribute. The second part of a CDN is a redirection architecture. Modern CDNs typically use DNS redirection [14], where the CDN operates DNS servers and the clients get redirected to the content server based on their DNS queries. For a comprehensive overview of CDNs and their performance evaluation, see [15].

Content distribution networks have also been used to transmission of live streaming content, but our focus in this paper is on using them for static, large files.

2.3 Multicast

The third content distribution mechanism we consider is multicast. Multicast is more aimed at delivering live and streaming content and not widely used for distributing large files, as our focus is. The reason for including multicast in our study is that it provides us with a convenient, intuitive lower bound on the network impact of any content distribution mechanism. We consider two forms of multicast.

Perfect multicast

In what we call *perfect multicast*, we assume there is one source placed in the network and that all the clients who are interested in the file join the multicast group at the same time. This means there will be only one multicast tree for the complete delivery of the file. Perfect multicast gives us a lower bound on the network impact of the distribution. We also assume that no packets are lost and no retransmissions are performed.

Realistic multicast

The assumption of all clients joining at the exactly same time is not realistic, hence we consider another multicast strategy, namely *realistic multicast*. Again, we have a single source placed in the network, but the clients join according to a specified arrival process. This means that the multicast group's membership changes dynamically, as does the distribution tree.

In realistic multicast, the source keeps on sending the file in cycles . We also assume there are no lost packets. Hence, if transmitting the file takes T seconds, a client which joins at time t will have the complete file time $t + T$.

We do not consider any reliable multicast techniques, since our two approaches above represent the baseline of what a multicast-based technology is able to achieve. In [13, 18] the authors present a reliable multicast scheme based on forward error correction with a return channel which has an overhead of about 10% in terms of bandwidth over standard multicast. Therefore, a reliable multicast scheme can be estimated to have a cost of about 10% more than our perfect and realistic multicast schemes.

2.4 BitTorrent

BitTorrent [6] is a new and popular form of peer-to-peer content distribution. It is especially used for delivering large files [12]. A BitTorrent network works as follows. Originally, the file to be distributed is available from one server, called *seed*. In addition to the seed, there is a *tracker* server which keeps track of all the clients in the network. A client who wants to download the file, contacts the tracker and receives a list of peers who are currently downloading that file or possess all of it.The client then picks some peers from this *peer set* and starts downloading chunks from them. BitTorrent uses a tit-for-tat policy, so that a client serves chunks to other peers who are serving chunks to it. A client will try to find the best set of peers from which to download, by trying different peers in the peer set. For a detailed description of how BitTorrent works, please see [12].

In this paper, we use BitTorrent as an example of a peer-to-peer content distribution system. Note that there are no guarantees that the peers a new client gets from the tracker are "good" (meaning offering fast downloads); discovering good peers in BitTorrent is the responsibility of the client.

2.5 Peer-Assisted Content Distribution Network

The last content distribution strategy we consider is a peer-assisted content distribution network. By this we mean a network based on a traditional CDN and using BitTorrent-like techniques for spreading the content from the content servers, using the peers as additional content sources. As opposed to BitTorrent where content is initially placed only in few (effectively random) locations at most, a peer-assisted CDN would seed the content through well-placed servers, just like in a CDN. But in addition to the normal CDN-like functionality, the peers would also be sources for downloading, just like in BitTorrent.

As our evaluation in this paper shows, a peer-assisted CDN provides a performance far superior to the other available content distribution mechanisms. Therefore we believe that despite the additional overhead, it proves that a content distribution system can greatly benefit from exploiting the peers in the network. However, as the comparison to the basic BitTorrent shows, this exploitation of the peers must be done with care.

3 Analytical Modeling

In this section, we will present an analytical model for comparing the different content distribution mechanisms. Our model is intended to show the similarities between the different mechanisms and to provide a basis for the experimental evaluation in Section 4. We will present three different models for BitTorrent, CDN, and multicast respectively, and discuss how they can be derived from each other.

Our cost model is based on the amount of traffic between the different autonomous systems (AS) in the network. We consider that inside an AS there is no additional cost for delivering the file and that the cost for transferring the file from one AS to another is proportional to the number of hops between the ASes and the size of the file. The number of inter-AS hops can also be used as a rough indicator of the available bandwidth and amount of loss, which directly correspond to the download time. Although this correlation is sometimes slightly weak (see [3, 19, 22] for evaluation and discussion about Internet paths), it can serve as a rough estimator for the download times experienced by the users. Our main goal, however, remains the evaluation of the network impact of the different content distribution mechanisms.

3.1 BitTorrent-Like Network

We start by deriving a cost model for a BitTorrent-like network. This applies to both the real BitTorrent and our peer-assisted CDN.

We consider a network where the nodes are autonomous systems (AS). We assume that there are I ASes in the network. The file we want to distribute is F bytes in size. We assume that we have initially N copies of the file seeded on some nodes. We assume that AS i has c_i clients who all want the file, so that the total number of clients is $J = \sum_{j=1}^{I} c_j$. We assume that the file is divided into K chunks, hence the size of a chunk is F/K bytes.

We will now calculate the cost for a client i to download the chunk k. We assume that the client is connected to some number of peers and that the client knows which chunks all these connected peers have (as per standard BitTorrent).

Let $Y_i(t)$ be a time-varying vector denoting the peers who are connected to peer i at time t, i.e.,

$$Y_i(t) = [y_j^i(t)], \tag{1}$$

where $y_j^i = 1$ is peer j is connected to peer i at time t and 0 otherwise. In the following we will ignore the time dependence and denote $Y_i(t) = Y_i$. This is because we consider only the downloading cost of *one chunk* and we make the simplifying assumption that the vector $Y_i(t)$ remains constant during the download of a single chunk. Note that for different chunks we would have different vectors $Y_i(t)$.

Let X_i be a (time-varying) matrix denoting the local view of the available chunks at peer i. That is,

$$X_i(t) = [x_{jk}^{(i)}(t)], \tag{2}$$

where $x_{jk}^{(i)}(t) = 1$ if peer j is connected to peer i ($y_j^i = 1$) *and* peer j has chunk k. Otherwise $x_{jk}^{(i)}(t) = 0$. Again, we simplify the notation by considering only one chunk and thus $X_i(t) = X_i$.

Let N_k^i be the set of peers connected to peer i who possess chunk k. The cardinality of this set can be calculated as

$$|N_k^i| = \sum_{l=1}^{I} x_{lk}^{(i)}. \tag{3}$$

Now, any of these $|N_k^i|$ peers can potentially upload chunk k to peer i. Uploading will happen if either peer i belongs to the list of "closest peers" of the uploading peer (i.e., peers which offer the best upload rate from the point of view of the uploading peer), or if the uploading peer happens to choose peer i by optimistic unchoking. Let $M_k^i \subseteq N_k^i$ be the set of peers which are connected to peer i, have chunk k, *and* have chosen to upload chunk k to peer i (either by normal uploading or by optimistic unchoking).

Therefore the cost for downloading chunk k is

$$C_k^i = \frac{F}{K} \min_{l \in M_k^i} (d_{il}(Y_i)), \tag{4}$$

where $d_{il}(Y_i)$ is the distance of the shortest path from peer i to peer l.

Equation (4) gives us the cost for downloading one chunk k. Because the set of possible uploaders M_k^i varies with time, we need to sum the costs for the individual chunks with their respective uploader sets M to get the total cost. The total cost for peer i to download the whole file is

$$C^i = \sum_{k=1}^{K} C_k^i \tag{5}$$

and the total cost for distributing the file to all the I peers is

$$\begin{aligned} C &= \sum_{i=1}^{I} C^i \\ &= \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{F}{K} \min_{l \in M_k^i} (d_{il}(Y_i)). \end{aligned} \tag{6}$$

Note that even though equation (6) does not explicitly include the time-variance of the download cost, this is included through the sets N_k^i and M_k^i which reflect the evolution of the system. We have assumed these sets to be constant for downloading a single chunk, but we explicitly allow them to change from one chunk to the next, hence our model does capture the essential properties of a BitTorrent-like content distribution system.

3.2 Content Distribution Network

In a CDN, files typically are distributed in whole, hence we have only one chunk $K = 1$. In addition, the CDN servers are placed in fixed locations and all clients know these. Therefore $X_i(t) = X$ for all ASes i and all times t. Therefore the matrix X simplifies to a vector of length I where element $X_i = 1$ is AS i contains a content server. Otherwise $X_i = 0$.

Theoretically, the set of possible sources of the file for client i includes the set of all content servers, i.e., the set M^i contains all the content servers ($M^i = M$), and the client is then redirected to one of them.

As in equation (4)– (6), the total cost for distributing the file to all the clients is then

$$C = \sum_{i=1}^{I} F \min_{l \in M}(d_{il}(X)) \qquad (7)$$

Note that the cost is analogous to the cost of downloading one chunk in BitTorrent (equation (4)) and can be obtained from the former equation by simply replacing the size of the file, the set of uploading peers M, and the placement matrix X with the equivalent expressions for a CDN.

3.3 Multicast

In the perfect multicast case, we have only one copy of the file ($N = 1$) and the vector X has only one element set to 1. The total cost in this case is then

$$C = F \sum_{i=1}^{I} d_i \qquad (8)$$

where $\sum_{i=1}^{I} d_i$ is the length of the multicast tree from the source node to all the client nodes. In the perfect multicast case, this is simply the length of the shortest spanning tree.

In the realistic multicast case, the multicast tree depends dynamically on the arrivals and departures of the peers. Let I' be the number of active ASes in the multicast group. Note that an AS will be in the multicast group if any client in that AS is downloading the file. Again, we assume that the multicast group remains stable for the sending of one chunk, but allow it to change between chunks.

Let the vector $Y_i = [y_j^i]$ denote the neighbors of AS i in the multicast group. In other words, $y_j^i = 1$ if AS j is the neighbor of AS i in the multicast group, and 0 otherwise. Therefore, the set of possible sources for the file for AS i is

$$M^i = \{j : y_j^i = 1\} \qquad (9)$$

The cost for one chunk in this case is then

$$C_k = \frac{F}{K} \sum_{i=1}^{I'} \min_{l \in M^i}(d_{il}(Y_i)), \qquad (10)$$

where $\sum_{i=1}^{I'} \min_{l \in M^i}(d_{il}(Y_i))$ is the length of the multicast tree from the source node to all the active nodes at that time. Summing equation (10) over all the chunks gives us the total cost for the realistic multicast case

$$C = \sum_{k=1}^{K} \sum_{i=1}^{I'} \frac{F}{K} \min_{l \in M^i}(d_{il}(Y_i)), \qquad (11)$$

This model can also be used in the perfect multicast case by setting $I' = I$ and ignoring time variance, in which case $\sum_{i=1}^{I'} \min_{l \in M^i}(d_{il}(Y_i))$ reduces to the length of the spanning tree $\sum_{i=1}^{I} d_i$ in (8).

Note that in contrast to the other mechanisms, we consider only multicast which starts from a single source. This may seem unfair to multicast, since it would of course be possible to have several multicast sources sending the same file, with each client joining the closest multicast tree. However, since the cost for multicast is basically the length of the spanning tree, this approach would simply create N subtrees in the AS graph, which would reduce the total cost by an amount proportional to $O(N)$. Given that the overall cost is proportional to the size of the network $O(I)$, the reduction of cost from multicasting from several sources would simply be a small factor in reducing this cost.

Recall that we assumed that IP multicast would be used. In the current Internet, IP multicast is not widely deployed and if we wanted to use multicast, we would have to use an application layer multicast approach. Since the topology of the overlay network in an application layer multicast does not correspond exactly to the topology of the underlying IP network, the cost would have to be increased by an additional factor, called *stretch*. Typical values for the stretch have been observed to be about 1.1 [4], although in some cases the authors in [4] report that the stretch can be quite significant (5 or possibly more). If we were to include reliable multicast in addition, we would expect another 10% increase in cost [13,18], giving a real, application-level multicast distribution about 20% additional cost compared to the values reported below for realistic multicast.

3.4 Comparison

If we compare equations (6), (7), and (11), we see that they are all of the same basic form

$$C = \sum_{ASes} \sum_{Chunks} size \times \min_{l \in M}(d(Y)),$$

that is, summing over all the chunks in the file, assuming that we can find the chunk in the best current location, which gives the cost for a given client to get the file. These individual clients are then summed together over all the ASes to give the total cost.

The only difference between them is how the set M of currently active peers is determined (and by extension, the set I' for multicast). Our focus for the remainder of the paper will therefore be around the dynamic properties of the different mechanisms, that is, how does the set M vary as peers join and leave the system.

3.5 Discussion on Cost

Our cost model takes into account only the number of inter-AS hops that we have to use to distribute a file to all the interested clients. Since our focus is on examining the impact of the different content distribution mechanism *on the network*, this choice is appropriate, because inter-AS traffic typically represents a larger cost to the ISPs than intra-AS traffic. Other metrics, such as analyzing the download time, have been studied in the literature [20].

Mechanisms which rely on seeding the network with an initial number of copies get a slight advantage in our cost model, since the cost of creating the initial replicas is not accounted for. This is the case for CDN, BitTorrent, and Peer-Assisted CDN. The reasons for this are three-fold. First, especially in CDNs, the content is replicated over a private network. Second, even if the replication was done over the Internet, the cost of replication is a one-time, fixed cost for a given file and number of copies. A file may be replicated 50 times, but if it is downloaded several 100,000 times, the cost of the initial replication is only a minor fraction of the total traffic. Since our focus is on popular content, we can ignore the cost of the replication. Furthermore, our results indicate that the performance of the Peer-Assisted CDN is almost independent of the number of initial copies, hence the replication cost for such a system can be kept low. Third, the cost would be the same for all the three mechanisms, hence including it would not make a difference in the relative performance.

4 Experimental Evaluation

For the evaluation of the different content distribution strategies, we performed several experiments in which we simulated the behavior of the different strategies with different parameters. In this section, we will explain the most important parameter choices.

We ran our simulations on AS-level network topologies which were generated by BRITE [16]. We used several different network topologies and report results for two different topologies in this paper. We used the built-in mechanisms for flat AS-level hierarchical topologies in BRITE. Here we report the results for 500 node topologies. In each AS we placed a number of clients which we varied as a parameter in our experiments to see the effects of file popularity. We used the same number of clients in each AS and varied the number from 1 to 5.

4.1 Server Placement

Each of the mechanisms requires us to place the initial copies (one for multicast, one or more for the others) in some locations in the network. We use the following placement algorithms. For BitTorrent, we create the initial seeds in random location, to reflect the current behavior of BitTorrent. For CDN and Peer-Assisted CDN, we use the greedy placement algorithm from [21] to determine the locations where the N initial copies are to be placed. This heuristic algorithm tries to place the servers in places so that the network traffic is minimized. For perfect multicast the placement does not matter, since the shortest spanning tree is unique. For realistic multicast, we place one copy with the same greedy algorithm as above.

4.2 Arrivals and Departures

Another key factor in our evaluation is the arrivals and departures of clients. For Perfect Multicast and CDN these do not matter, since Perfect Multicast assumes that all clients arrive at exactly the same time and in CDNs, every client needs to download the file by itself and the amount of traffic generated is thus independent of the arrivals of the clients.

For BitTorrent, Peer-Assisted CDN, and Realistic Multicast, client arrivals and departures play a crucial role in determining the cost of content distribution. We use the following mechanism for determining when clients arrive and when they depart the system. We make the simplifying assumption that a client will not depart until the download is completed. Although the assumption about clients downloading the complete file may sometimes be too restrictive, our results in the next section show that the performance is not very closely tied to the departures of the clients. This, in turn, implies that even if clients depart early, the effect on the overall cost of the distribution is likely to remain small. Therefore we believe that this simplifying assumption of complete downloads is justified.

We use a discrete-event simulator for our experiments. In the simulator, we have divided the time into *rounds* and during one round, we assume that we are able to transfer one small piece of the file (e.g., one chunk in BitTorrent). At the beginning of a round, we assume that a random number of new clients arrive. The number is drawn from a uniform distribution between 0 and A. The locations where the new clients arrive are randomly determined. At the end of a round, we draw a uniformly distributed random number between 0 and D to determine the number of clients that can leave the system at the end of that round. Note that because of our above assumption that all clients download the complete file, the actual number of departing clients is the smaller of random number and the number of clients who possess the complete file. Although these arrivals and departures produce session times which do not always correspond to values observed in the real world, the advantage of using synthetic arrival traces over real-world traces is that synthetic traces allow us much more flexibility in evaluating the sensitivity of the content distribution mechanisms to client behavior. As already evident in the model (see Section 3.4), client dynamics are at the key component in the total cost, hence it is important to be able to evaluate the sensitivity of the mechanisms with as many different arrival and departure rates as possible.

In the following, when we talk about client arrivals and departures, we refer to the values of A and D. For example, "arrival 40 and departure 20" means that $A = 40$ and $D = 20$, i.e., at the beginning of a round, 0–40 clients arrive, and at the end of a round, 0–20 clients depart (assuming they have the complete file). The justification behind the choice of these arrival and departure rates is that it allows us to investigate different arrival/departure ratios, as well as different absolute values for arrivals and departures.

5 Results

We will now present the results from our simulation experiments. We present the results for 6 different sets of parameters (number of nodes and client arrivals and departures). We ran the experiments for a much larger set of parameters and the results we obtained were similar to the ones reported here. Figure 1 shows the results for a network of 500 nodes (as described in Section 4) and 4 different combinations of client arrival and departure rates. Because of space limitations, we do not show other results.

In each of the plots, the x-axis is the number of copies (for CDN) or the number of initial seeds (for BitTorrent and peer-assisted CDN), that we create at the beginning of the simulation. We ran our simulations for the full range (up to 500 or 1000 initial copies), but only report the results for up to 100 initial copies. This is because it is unrealistic to assume that we would be able to seed a significant fraction of the total network nodes with a copy. Note that in our simulations, 100 nodes represent 10 or 20% of the total nodes. Gao reports that there were 6474 ASes in use in January 2000 [10] and Akamai states on their website that they currently have CDN servers in about 1100 ASes [2]. This represents the extent of a large CDN which corresponds to our choice of number of CDN servers or seeds that we report in the figures.

On the y-axis, we plot the average download cost in hops *per client*. This represents the number of inter-AS hops that an average client needs to cross to get the file. For CDN, this is the average distance to the closest CDN server. For Realistic Multicast, this is the average length of the multicast tree and for Perfect Multicast this is simply the length of the spanning tree divided by the number of nodes, i.e., $(I - 1)/I$ in our case. BitTorrent and Peer-Assisted CDN are based on downloading small chunks from the file from several sources, and we count the cost for each chunk individually, and by weighing this cost by the size of the chunk, we get the average cost for the complete file.

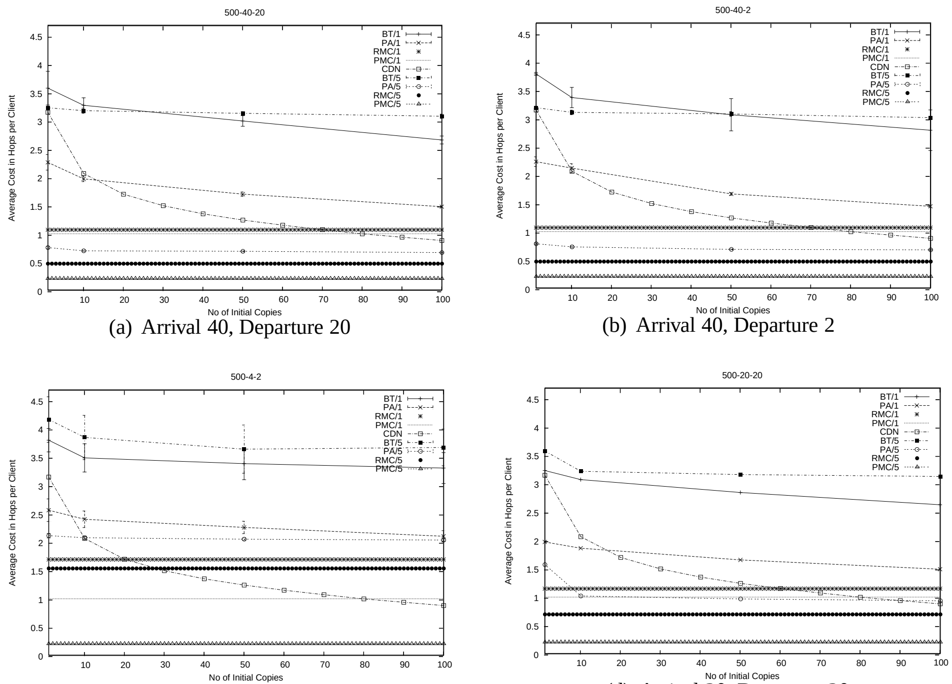
We ran each experiment in the figures multiple times and the figures report the average values. Some figures also plot the intervals between the minimum cost and maximum cost observed in the experiments. In most cases we observed that the differences were minimal. Perfect Multicast and CDN are independent of the client arrivals and their results can be calculated using equations (7) and (8).

The results can also be used to estimate how much additional load new clients would bring. Since the cost on the y-axis is calculated per client, any new client would have, on average, the given cost for downloading the file.

Each graph shows several lines, which are named `strategy/c`, where `strategy` is the name of the strategy (BT is BitTorrent, PA is Peer-Assisted CDN, CDN is CDN, PMC is perfect multicast, and RMC is realistic multicast) and `c` is the number of clients per AS (for the strategies where it matters).

Figure 1 shows the results for a 500-node network. We show 5 different arrival and departure rate combinations. In 3 of the cases, the ratio of arrivals to departures is the same (i.e., 2) and in one of them (Figure 1(b)) we show a larger ratio. In Figure 1(d) we show a case where the ratio is 1. Figures 1(a) and 1(c) show that for the same arrival/departure ratio, an increase in the absolute arrival and departure rates leads to improved performance for Peer-Assisted CDN and Realistic Multicast. We will discuss the implications of this in Section 6. Comparing Figure 1(d) with the higher arrival to departure ratios, we see that the results are also similar, which indicates that the systems are not very dependent on the departures of the clients, i.e., frequent client departures do not hurt the performance.

All in all, we can make the general observation that BitTorrent has the highest cost of all strategies, followed by



(a) Arrival 40, Departure 20
(b) Arrival 40, Departure 2
(c) Arrival 4, Departure 2
(d) Arrival 20, Departure 20

Figure 1. Results for 500 nodes in network

CDN and Peer-Assisted CDN (in either order). Both multicast strategies, Realistic Multicast and Perfect Multicast, typically have a lower cost. Since Perfect Multicast represents the lower bound, its good performance is not surprising. Between Realistic Multicast, CDN, and Peer-Assisted CDN, the difference is typically not very large. Recall that the Realistic Multicast is based on several simplifying assumptions (IP level multicast, no packet loss), so that in reality, the costs would likely be roughly equal.

In some parts of the parameter space CDN has a lower cost than Peer-Assisted CDN and vice versa. As the number of CDN servers or seeds in Peer-Assisted CDN increases, the cost of both of the mechanisms tends to zero, since eventually every AS would have a CDN server or a seed. As mentioned above, the graphs present a realistic design space of a CDN and further study is necessary to evaluate the differences of CDNs and the Peer-Assisted CDN and determine the causes of these differences. Recall that a CDN would typically imply other costs, such as managing the private network for replicating content which cannot always be measured in terms of network hops in the Internet (see Section 3.5).

Recall that the standard unicast distribution from one server is roughly equivalent to a CDN with only 1 server. Figure 1 allows us to make the interesting observation that the standard BitTorrent imposes a *higher* cost on the network than each client downloading the file from a single server on its own. On the other hand, BitTorrent does have the advantage that it spreads the load over all the peers, not just a single server as in the unicast case. Nonetheless, from a purely network traffic point of view, BitTorrent seems extremely wasteful of resources.

6 Discussion

We will now discuss the implications of our results on content distribution systems. As stated above, the typical ranking of the different strategies is, from worst to best, BitTorrent, CDN, Peer-Assisted CDN, and Multicast, where the relative performance of CDN and Peer-Assisted CDN varies. This conclusion holds for all the results we show here, as well as for the results we have omitted.

Especially for a large number of clients per AS, the results for the Peer-Assisted CDN are encouraging, as its cost is very low.

6.1 Difference between BitTorrent and Peer-Assisted CDN

The difference between BitTorrent and the Peer-Assisted CDN is striking, considering that they both use basically the same system for distributing the content. The difference lies in the implementation of the Peer-Assisted CDN which aims at optimizing the performance at every step.

BitTorrent gives no guarantees about the peers on the list given by the tracker; these could be peers that are very far from the downloading peer. Even though the peers try to discover nearby peers, there are no guarantees that they will find them. This is a large factor in the high cost of BitTorrent, and explains why in some cases an increased number of clients per AS increases the cost. See Section 6.3 for an explanation on this phenomenon.

In contrast, the Peer-Assisted CDN always gives a list of "best" (i.e., closest) peers from the tracker. Even though the peer might actually not be able to download from the absolute best peers, it will try to download from the peers on this list, which are all very good candidates for download.

Another observation we can make is that in the Peer-Assisted CDN, the number of seeds does not influence the performance much, as opposed to BitTorrent. This means that *selecting good neighbors* from which to download is the key to peer-assisted content distribution mechanisms. Other factors, such as number of initial copies, size of the network, or peer behavior patterns, have only negligible effect on the overall performance.

6.2 Effect of Arrivals and Departures

The effects of the client arrivals and departures (i.e., client dynamics) are clearly visible in the results. The strategies which especially depend on this are Peer-Assisted CDN and Realistic Multicast. Also BitTorrent reacts to the client dynamics, however, the effects are typically less pronounced.

As Figure 1(d) shows, even when the "arrival" to "departure" ratio is low, the performance of the mechanisms does not degrade markedly. However, it is better to have more arrivals than departures to keep enough sources for chunks in the system. Typically, we observed that an arrival to departure ratio of 2 is sufficient. However, as shown by the comparison of Figure 1(d) and 1(c), having a high enough absolute arrival rate already guarantees good performance. Therefore, we can conclude that as long as clients are arriving sufficiently frequently, their departures have little effect on the overall performance.

This is a very promising result, since efficient content distribution strategies are extremely vital for *popular content*, which many clients want to download simultaneously. In these cases, we would have a situation where many clients arrive, i.e., similar to the case in Figures 1(a) and 1(b).

Also encouraging is the comparison of Figures 1(a) and 1(b). Even though Figure 1(a) has a departure rate an order of magnitude larger than in Figure 1(b), the actual difference in cost is quite small. This confirms our result from above, that if the arrival rate is high enough (i.e., content is popular enough), then it does *not* matter whether the clients

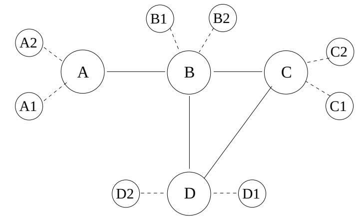


Figure 2. Example with different number of clients per AS

remain as seeds after they have completed their downloads. Hence, the performance is not tied to altruistic users, but can also support a relatively large number of free-riders without a negative effect on performance.

6.3 Number of Clients

As expected, systems, where client dynamics play a large role, benefit from a larger number of clients per AS, since this makes the dynamics on the inter-AS level much more stable. This applies to both Peer-Assisted CDN and Realistic Multicast. For the normal BitTorrent however, increase in the number of clients per AS typically increases the overall cost. The following example illustrates why this can happen.

Consider the network topologies shown in Figure 2. The network has 4 ASes (A, B, C, and D) and each AS has two clients as denoted by the small oval. First, consider the case where each AS has only 1 client, i.e., only clients $A1$, $B1$, $C1$, and $D1$ are in the network.

We assume that each of the links between the ASes has a cost of 1. Furthermore, assume that ASes B and C have a copy of the file, in other words, $B1$ and $C1$ are the seeds in the system. We assume that clients are coming randomly in the network and that arrival rate is 2 and departure rate is 1 and tracker will give 2 random peers when a client joins the network. We assume that there is only 1 chunk to be transmitted. We also assume that peer will upload to the closest client at a time.

Consider the cost of the client $A1$. When client $A1$ joins the network, it will get two peers from the tracker. Since there are only two peers available, $B1$, and $C1$, $A1$ will get both of them and pick the better of those two, i.e., $B1$. So the expected cost of A1 to download the file is 1 hop. Note that even though the tracker gives a list of random peers, a downloading client even in standard BitTorrent will pick the best peer from the list it gets, i.e., $B1$.

Let us now consider the case where all the clients in Figure 2 are in the network and calculate the expected cost for client $A1$, under the same assumptions about the arrivals and departures. Note that we also assume that the seeds $B1$ and $C1$ remain in the system.

There are now 6 clients ($A1$, $A2$, $B2$, $C2$, $D1$, and $D2$) who want to join the system and 2 clients ($B1$, $C1$) are the initial seeds. Since 2 clients arrive each round and the arrivals are randomly distributed, client $A1$ can arrive in the first, second, or third round.

If $A1$ comes in the first round, we have the same situation as in the 1 client case and the expected cost is 1.

If $A1$ comes in the second round, there are 10 possible combinations of clients having arrived in the first round, i.e., ($A2$, $D1$), ($A2$, $D2$), ($A2$, $B2$), ($A2$, $C2$), ($D1$, $D2$), ($D1$, $B2$), ($D1$, $C2$), ($D2$, $B2$), ($D2$, $C2$), ($B2$, $C2$) and one of the two arrived clients will have left the system, since it has completed the download and the departure rate was assumed to be 1. The expected cost in this case is $\frac{5}{6}$. (See [9] for details.)

In the case that $A1$ comes in the third round, the expected cost can be shown to be $\frac{21}{20}$ hops.

Since client $A1$ can arrive in any of these rounds (with equal probability), the overall expected cost is then $1/3(1 + \frac{13}{12} + \frac{21}{20}) = 1.044$ hops.

This calculation explains why the cost of BitTorrent can increase as the number of clients per AS increases. As the calculation shows, the amount of increase depends on the arrival and departure rates, as well as on the number of chunks to be distributed. Evaluating the inherent sensitivity of BitTorrent to this phenomenon is part of our future work.

The Peer-Assisted CDN is immune to this problem, since the tracker always gives the list of closest peers to the client. In other words, in the above case of peers $A2$, $B1$, and $C1$ being in the system when $A1$ arrives, the list from the tracker would always contain $A2$ and $B1$, thus resulting in a cost of zero hops.

6.4 Implementation Concerns

Given the good performance of the Peer-Assisted CDN, the logical follow-up question is how hard is it to implement such a scheme? As our results show, the number of initial seeds is of little importance, as the performance is relatively stable across a wide range of initial seeds. Therefore, we do not require an extensive network of well-placed content servers as in a CDN, although such a network could be used.

The second big problem relates to how the tracker can give the list of the best peers to new clients who join the system. Modern CDNs achieve this functionality through their vast network of content and redirection servers, which demonstrates that such a system can be built. Other proposals for systems which are able to determine the “distances” (e.g., latency) between hosts in Internet include Vivaldi [8], PIC [7], and NPS [17]. Any such system could serve us as a means to allow the tracker to determine which clients are closest to the new peer. Some such systems, e.g., Vivaldi [8], piggyback their measurements on existing application-level traffic, making the overhead of such a system manageable.

Therefore the implementation of a peer-assisted content distribution system does not require the deployment of a large infrastructure, nor does it necessarily place undue burden on the tracker.

7 Related Work

Krishnamurthy et al. [15] have studied the performance of several existing CDNs. Their focus is on examining the amount of content served by different CDNs and evaluating the type of content served by CDNs. They also perform an extensive evaluation of the performance of the different CDNs, in terms of download latency and redirection performance. In contrast, our work in this paper focuses on evaluating the network impact of CDNs, relative to other possible content distribution strategies.

Biersack et al. [5] discuss the performance of peer-to-peer networks in file distribution. They look at three different types of content distribution schemes, linear, tree, and forest, and derive an expression for the upper bound for the time which it takes until all peers have completed the download of a file. Their results also validate the usefulness and scalability of peer-to-peer content distribution technologies, but they do not compare peer-to-peer technologies with existing content distribution mechanisms. Our focus is on evaluating the impact of different content distribution technologies, including but not limited to peer-to-peer technologies, and gain insight into how a practical peer-to-peer content distribution system should be built.

Qiu and Srikant [20] present a model for BitTorrent-like networks for evaluating the evolution of the number of nodes in the network as well as the download times. They model the network as a fluid model and derive expressions for the average number of seeds and the average download time of a chunk. They also discuss the incentive mechanisms in BitTorrent and present strategies for peers to select to which peers they should upload. They do not address the problems of locality in the downloads and the resulting wide area traffic, which is the focus of our study in this paper.

Gkantsidis and Rodriguez present a network coding based approach for distributing large scale content [11]. Their approach has some similarities to our Peer-Assisted CDN mechanism. In the network coding system, there is one source for the file and several nodes which are interested in it. The source sends out the file in chunks and the nodes are able to generate new chunks through network coding [1]. They present an architecture for spreading the file through network coding and evaluate the performance of their architecture. The evaluation focuses on the download time and number of concurrent users. The similarities to our Peer-Assisted CDN are that both systems exploit the peers to distribute the files. However, in our work, we explicitly consider the network impact of the content distribution and perform a comparison between different content distribution strategies. As the comparison between BitTorrent and the Peer-Assisted CDN indicate, a peer-based content distribution system can have a wide range of performance values and such systems need to take much care in deciding how to exploit the peers.

8 Conclusion

In this paper we have evaluated several different content distribution mechanisms in terms of how much wide-area network traffic they cause. The models considered were unicast, standard CDN, two different multicast strategies, a BitTorrent-like peer-to-peer network and a Peer-Assisted CDN, which exploits the proximity of peers. We have developed an analytical model for the different content distribution mechanisms and our model shows that the differences between the mechanisms stem mainly from the dynamics of the system.

Our experimental results highlight the following conclusions. First, exploiting peers to distribute content can be highly beneficial, as demonstrated by the good performance of the Peer-Assisted CDN. Second, this exploitation of peers must be done carefully, maximizing the benefits at every step. This is clear from the large difference between BitTorrent, which exploits the peers naively, and Peer-Assisted CDN where we have optimized every step.Third, for highly popular content, a peer-assisted strategy does not need to rely on altruistic users, but can support a large number of free-riders. Fourth, a carefully planned peer-assisted content distribution has performance comparable to a realistic multicast scheme.

Our future work will include an analysis of the sensitivity of the different mechanism to the choices of the different parameters. In addition, we will evaluate different algorithms for determining which peers are closest to a given peer, since selecting good download sources was seen to be the main factor affecting performance.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] Akamai.

[3] A. Akella, S. Seshan, and A. Shaikh. An empirical evaluation of wide-area Internet bottlenecks. In *Proceedings of Internet Measurement Conference*, Miami Beach, FL, Oct. 2003.

[4] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002.

[5] E. W. Biersack, P. Rodriguez, and P. Felber. Performance analysis of peer-to-peer networks for file distribution. In *Proceedings of Fifth International Workshop on Quality of Future Internet Services*, Barcelona, Spain, Sept. 2004.

[6] BitTorrent web site.

[7] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet coordinates for distance estimation. In *Proceedings of International Conference on Distributed Systems*, Tokyo, Japan, Mar. 2004.

[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of ACM SIGCOMM*, Portland, OR, Aug. 2004.

[9] S. Das. Evaluation of network impact of content distribution mechanisms. Master's thesis, Darmstadt University of Technology and Indian Institute of Technology Kharagpur, May 2005.

[10] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, Dec. 2001.

[11] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In *Proceedings of IEEE Infocom*, Miami, FL, Mar. 2005.

[12] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A. Al Hamra, and L. Garcés-Erice. Dissecting BitTorrent: Five months in a torrent's lifetime. In *Proceedings of Passive and Active Measurements*, Apr. 2004.

[13] M. Jung, J. Nonnenmacher, and E. W. Biersack. Reliable multicast via satellite: Uni-directional vs. bi-directional communication. In *Proceedings of KiVS'99*, Darmstadt, Germany, Mar. 1999.

[14] J. Kangasharju, K. W. Ross, and J. W. Roberts. Performance evaluation of redirection schemes in content distribution networks. In *Proceedings of 5th Web Caching and Content Distribution Workshop*, Lisbon, Portugal, May 22–24, 2000.

[15] B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. In *Proceedings of Internet Measurement Workshop*, San Francisco, CA, Nov. 1–2, 2001.

[16] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *Proceedings of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS*, Cincinnati, OH, Aug. 2001.

[17] T. E. Ng and H. Zhang. A network positioning system for the Internet. In *Proceedings of USENIX*, June 2004.

[18] J. Nonnenmacher, E. W. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast. *IEEE/ACM Transactions on Networking*, 6(4):349–361, Aug. 1998.

[19] V. Padmanabhan, L. Qui, and H. J. Wang. Server-based inference of internet link lossiness. In *Proceedings of IEEE Infocom*, San Francisco, CA, Mar. 2003.

[20] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *Proceedings of ACM SIGCOMM*, Portland, OR, Sept. 2004.

[21] L. Qiu, V. N. Padmanabhan, and G. M. Voelker. On the placement of web server replicas. In *Proceedings of IEEE Infocom*, Anchorage, AK, Apr. 22–26, 2001.

[22] A. Zeitoun, C.-N. Chuah, S. Bhattacharyya, and C. Diot. An AS-level study of Internet path delay characteristics. In *Proceedings of IEEE Globecom*, Dallas, TX, Nov. 2004.