Dear Dr. Fonseca,

we thank you and the referees for reviewing our paper. Below we provide detailed answers to the reviewers' comments. Our responses are marked by (*). We highlight the changes in the manuscript using blue italic font.

Best regards,
Amit Mondal
Ionut Trestian
Zhen Qin
Aleksandar Kuzmanovic

Reviewer #1:

The paper introduces the interesting idea of combining P2P and CDN techniques to address the problem of content availability and low performance that affects many of the current BitTorrent swarms. The authors show that their strategy can be deployed in an incremental way, and that the gains are substantial even if a small fraction of users deploys the proposal.

General questions (please, address them in the paper, in case they are not already addressed)

1) is the proposed method distributed? or does it require a coordinator? does the MirrorPlane rely on a central node?

(*)
There are two points to be made here:

- First, MirrorPlane is deployed by individual ISPs and it monitors swarm performance in the local ISP. We have shown in the paper, with a back of the envelope calculation, that MirrorPlane can easily scale to handle the traffic and load generated by millions of p2p users (page 19, 4th and 5th paragraphs of section 6, now highlighted).

- Second, there is no reason why MirrorPlane cannot be enhanced by incorporating distributed system characteristics. For example, MirrorPlane can also be replicated to ensure fault tolerance and avoid having a single point of failure. The idea can also be extended to design a distributed hierarchical version of MirrorPlane. In the distributed version, a set of regional MirrorPlane nodes monitor local swarm performance and report collected information to a centralized location that makes the replication decision. The replication decision is passed to the p2p clients through the intermediate MirrorPlane nodes.

Finally, coming out with an entirely distributed MirrorPlane was not the main goal of the paper but merely to show how unused peer bandwidth can help reduce inter-AS traffic volumes.
---

2) megaupload.com was recently closed. would it be easy to close a system like yours?

(*) Policies can be implemented in the replication part of MirrorPlane to avoid the active replication of copyrighted content therefore easily avoiding any copyright related issues.

Note also that these policies are implemented only on the metadata associated with the file. In file sharing systems, metadata (which is usually found either directly in the filename or on forums where content is shared) tells users what the file contains. Indeed, it would be very hard to go at the content level to determine file specifics. This is a problem with p2p systems in general and not with MirrorPlane. One can imagine how a malicious user of a p2p system can just rename a child pornography video as a popular movie and insert it in a p2p system. It is the job of the users of the p2p system to tag it (thus producing more metadata) so that other users do not get tricked.

We have added the following paragraph in section 6 (page 20, paragraph 9 of section 6, security subsection):

"Note that the content filtering part of MirrorPlane operates on the metadata associated with the file. This comprises the filename and other information that users associate with the file on torrent websites. It would be very hard to go at the content level to determine file specifics. This is a problem with p2p systems in general and not with MirrorPlane. Indeed, a malicious user of a p2p system can just rename a child pornography video as a popular movie and insert it in a p2p system. This problem is usually avoided by users providing feedback about the file."

However, again, our research is not directed towards designing such policies as we mainly aim to show how unused upload bandwidth can help reduce inter-AS traffic volumes.
---

3) what are the key shortcomings of your method?

(*) If there is a sudden change in the popularity of swarms, our method might not be very effective at the beginning as it will not be able to create enough replicas of popular content quickly and therefore leechers will end up downloading content from remote peers. This generates inter-AS traffic. However, once the content is available in the local AS, generating more inter-AS traffic is avoided.

We have added the following paragraph at the end of the Discussion section (page 20, paragraph 10 of section 6).

"Flash Crowds. In case of a sudden change in the popularity of swarms (for example when long awaited content such as a TV show episode is added) our method might not be very effective in the beginning. This is because creating replicas of popular content takes some time. This will generate some amount of inter-AS traffic (by users who would normally download the content because of interest). However, once the content is sufficiently available in the

local AS, generating more inter-AS traffic is avoided."
---

4) you claim that your system can filter content such as child pornography. How so? What if the content is encrypted?

(*) Every swarm has a unique ID, no matter if the content is encrypted or not. MirrorPlane has a list of illegal swarms and avoids the active replication of such content.

As we stressed above, the information we have about the content in each swarm comes from the metadata associated with that swarm. This either comes from the filename or from what users post on forums or on tracking websites about it. We stress again that the metadata associated with the content being wrong is a problem in most p2p systems.
---

5) if a user starts downloading content that he has not requested, will he share that content even before becoming a seed?

(*) This is a built in feature of the protocol. As soon as the user downloads a content block, it can automatically send that block into the swarm.

We have added the following phrase in section 3.1 (page 9) to address this:

"Note that, as soon as the user downloads a content block it can provide it back to the swarm as usual in a p2p system."

Knit picks

page 4, last para section 2.1, second CDN => first CDN
page 19, type of attach => type of attack

(*) We have changed this and we have marked it in the paper.

Reviewer #2:

This paper proposes a CDN-based content delivery approach for p2p systems. By appropriately replicating content and using existing altruistic behavior, it is possible to significantly decrease inter-AS traffic and at the same time improve performance.

The paper looks at an important problem and is well-written. I think it would be improved by addressing the following issues:

(1) Piatek et al. (2009) find that optimizing for locality alone may degrade performance. Because of this, in their experiments they swap distant for local peers only if the switch does not degrade performance (because otherwise a peer would not follow the protocol). On the contrary, the authors find that P2P-

CDN both improves performance (Figure 9) and reduces inter-AS traffic. Is it the case that performance only improves on aggregate (Figure 9 only shows the cdf) but not for every particular user? The authors could discuss this and whether a user would sometimes be better off downloading from distant peers.

(*) In our approach, we actively replicate content according to its popularity to harness excess upload bandwidth left unused in unpopular swarms. As a result, our approach improves performance for popular swarms without affecting the performance of unpopular ones. In our approach, we are basically effectively optimizing the use of the available upload resources in an ISP.

As mentioned in the paper (now highlighted) on page 10 in subsection 3.1.3 last paragraph:

"One final policy that we deploy is to give a strict priority to the original torrent, e.g., P, over the replica torrent, e.g., R."

and on page 15 in subsection 4.2.4 end we added and highlighted:

"Figure 9(b) shows that both P2P-CDN and AS-biased approaches have similar CDF of download times for the unpopular torrents. In the P2P-CDN case, the performance is not degraded despite the replication process. This is because uploads towards original torrents are given a strict priority over replicas (Section 3.1.3). Hence only excess, still plentiful, resources are utilized for replication. In summary, P2P-CDN approach speeds up download times of the popular torrents, and it achieves this without hurting the download times of the unpopular torrents. Therefore, in our approach, we actively replicate content according to its popularity to harness excess upload bandwidth left unused in unpopular swarms. We discuss below how the performance of unpopular torrents could be further improved."
---

(2) Users could exhibit less altruism in a setting with content replication. Currently, a user may download a file because he wants it and then doesn't mind seeding (once he is done downloading it). But a user may not be willing to do that for a file that he is not interested in having. How altruistic a person is significantly depends on the context, so it should not be taken for granted that a person will exhibit the same level of altruism when the context changes.

(*) Our belief is that most altruism comes from lazy or negligent users and not because someone intentionally wants to share content. Thus, we do not think this is going to change even in an active replication setting. However we agree with the reviewer that this is a matter of speculation and hence we added the following phrases on page 19, paragraph 7 of section 6:

"How altruistic a person is significantly depends on the context, so it should not be taken for granted that a person will exhibit the same level of altruism when the context changes. However, there is no reason why we would suspect that users would dramatically change their behavior in the near future."
---

(3) The content replication creates a certain overhead (because peers end up downloading files they are not interested in having). What percentage of inter-AS traffic (e.g., in Figure 7) is overhead? This issue does not arise when the peer already has the corresponding torrent (from a previous download), as the authors note at the end of page 8. How often is this the case?

(*) When actively replicating content, peers download content only from local peers, thus does they do not generate any more inter-AS traffic.

Refer to point 1 of the same reviewer.

Other comments:

Page 6, Figure 3: Why is there an increasing (resp. decreasing) trend at the beginning (resp. end) of the period?

(*) The increasing and decreasing trends at the beginning and end correspond to the warm-up and wrap-up periods in our experiment. What happens is that at the beginning we start exploring peers through the tracker and after that, through the PEX mechanism we discover more and more peers. At the end we stop getting new peers through PEX and only monitor the ones we already know.

Page 7, 1st paragraph: "the aggregate upload bandwidth. even when we account for the asymmetry of upload and download links". Does Figure 4 account for this asymmetry? I don't see how a reasonable comparison can be made without accounting for this asymmetry.

(*) Yes, Figure 4 accounts for the asymmetry of the upload and download links. We have included the following phrase on page 7 at the beginning of subsection 2.4 to address this:

"Note that in Figure 4 we account for the asymmetry of upload and download links."

Algorithm 3.1: The case that aggrUpload is between aggrDownload + 2*stdev and aggrDownload + 3*stdev is not included in the algorithm. To me, it seems like a typo.

(*) Our algorithm can basically be described as follows:

1) AggrUp < AggrDown => we do not actively replicate as upload is already scare therefore we do not want to increase load.

2) AggrDown < AggrUp < AggrDown + 2*stddev => increasing popularity so we replicate content actively, enqueue in the todo replicate list with weight = 1/rank.

3) AggrDown + 2*stdev < AggrUp < AggrDown + 3* stddev => enough upload

capacity for the swarm, do not do anything, but do not take resources from this swarm, otherwise it will move to (2) and we have to actively replicate it.

4) AggrDown + 3*stddev < AggrUp => way more upload capacity than download, probably it will never be utilized and needed. Take away upload b/w from this swarm and allocate to the swarm in the queue that we build in (2) with highest weight.

The reviewer is referring to point 3 from above which was not included in the algorithm description from the paper because no action is taken in that case.

Reviewer #3:

This paper presents an approach to p2p content distribution that actively replicates content among peers in order to reduce inter-AS traffic. The intention is to show that p2p need not work against the interest of ISPs, and, in fact, can improve the download speeds of peers as a side effect because content is closer to them in the network. The authors show that, contrary to popular belief, BitTorrent swarms generally have enough upload capacity to satisfy download demands; the problem is that most swarms today do not efficiently distribute content to make the upload bandwidth available or useful to peers or the system as a whole.

This paper is partway there. You provide some interesting observations backed by experiments, and you propose policies that attempt to replicate content to make the distribution of content among swarms generally more desirable. The problem is you never crisply state either your goal or your solution. Parts of each are scattered throughout the paper, but even then, I think there are pieces missing.

(*) Our goal is to design a p2p file sharing model that minimizes inter-AS traffic without affecting the performance observed by peers.

We now purposely state this goal in the introduction (page 2, end of paragraph 4 of the Introduction):

"Our main goal is to reduce inter-AS traffic volumes without sacrificing peer performance measured by download time."

GOAL

Take the goal of the paper. Your goal, vaguely stated, is to reduce inter-AS bandwidth. But that can't be all of it--you also want to ensure that individual peers see faster download times, or at least it seems that way used on your results. Or is it stronger than that--is the goal that no peer's download speed decreases? One purpose of the result might be to show that it's always possible to replicate content such that both are true: the inter-AS bandwidth necessarily goes down without sacrificing any swarm's ability to distribute its own content. But the paper never states that explicitly, and I don't know whether that's even true. Instead, the goal seems to be stated in such a way that the results "back it up" by showing that, indeed, things generally get "better".

I think what the paper needs is a way to measure "better". Define a metric. Choose what you're willing to trade off for what, or at least explicitly prioritize decreasing inter-AS bandwidth, increasing average peer

bandwidth, increasing worst-case peer bandwidth, etc.

(*) Our goal is to cut down inter-AS traffic without affecting the download times of individual peers. Basically to summarize it, we want to show how unused upload bandwidth can help reduce inter-AS traffic volumes.

As we stated in a previous response to reviewer 2, we strictly prioritize traffic downloads that a peer was actually interested in over replicated torrent traffic so as not to decrease current peer performance. Basically the download times in our system are always improved (note Figure 9). Therefore the metric that we further analyze in the paper is the inter-AS traffic.

SOLUTION

I didn't know you had built the MirrorPlane system until too late in the paper. In fact, there wasn't mention of a system at all until page 10, and even then, I can't tell whether you built a simulator or something that can be deployed in the wild. Building a system is a big deal, so flaunt it! Your abstract should at least mention that you have formulated a solution well enough that you were able to implement it. "We built a system called MirrorPlane that..." Until page 10, you allude to "mechanisms", but I don't know what that means exactly.

(*) Our implementation is a proof-of-concept prototype and is described in section 5 and an overview is provided in Figure 11.

To summarize, first we have implemented the monitoring module. It was implemented by modifying the bitflu BitTorrent client. It basically monitors the swarms inside an ISP and reports monitoring information to the replication scheduling module.

Second, the replication scheduling module is currently implemented in an event-driven simulator where we replay our collected traces and hence validate our solution.

In terms of the approach, I like that you have a policies section that outlines how replication occurs. I think it would be stronger if you separated it into two sections: one that explains, at a high level, the where, when, what of replication, and another that distills the actual policies and defines them more concretely. As written, I have trouble deciphering exactly how replication happens. You say that content is replicated to "peers that have finished downloading a torrent and have excess resources." Does that mean you ask peers to download content that they don't want? If so, that means that more content has to be distributed to peers (for the purpose of allowing those peers to upload content that they never wanted in the first place); how does that affect the results?

(*) First, the way we see section 3.1 (Replication Policies) and section 3.2 (Replication Mechanisms) is exactly what the reviewer suggests. section 3.2

introduces what to replicate (the most popular torrents), when to replicate (depending on demand), and where to replicate (users that have finished downloads and have excess resources). The specifics come in section 3.2 where we introduce algorithm 3.1 that puts all these together. To explain this we have added the following paragraph on page 9 in section 3.1:

"Below, we explain at a high level the where, when, and what of the replication process. In Section 3.2 we introduce Algorithm 3.1 and also describe more specifically how replication takes place."

Second, our solution indeed pushes content to underutilized peers so that they can contribute their excess upload bandwidth to the swarms where it will be effectively utilized. This is already incorporated in Figures 9 & 10 in the paper. So yes, it means that we are distributing content that peers do not want but this content comes from the local AS so as not to generate inter-AS traffic.

Finally, I'm missing key points related to MirrorPlane. Who actually does the controlling? The scheduler or the "replication coordinator" (which comes out of nowhere on page 19). Is it a centralized server? A distributed set of MirrorPlane peers? What protocol do the peers use to talk to this component?

(*) In order to make things clearer we have included a new figure in the paper (Figure 11 at the moment) in section 5. Also we have pointed out all around section 5 how the different components inter-work and how the different actions are triggered.

By coordinator we mean scheduler (a term we have used in most of the paper) and we have replaced this in the sections involved in an effort to be consistent across the paper.

Related work: the authors of Antfarm have follow-up work called V-Formation in SIGCOMM 2011.

(*)
We have included the following paragraph in the related work section (page 22, paragraph 7 of the related work section):

"Peterson and Gun Sirer follow up with further work in this area similar in spirit to AntFarm. In [46] they introduce V-Formation and the Content Propagation Metric, which enables content distribution systems to make efficient use of bandwidth from all sources. Again the main difference between our work and V-Formation is that they optimize bandwidth usage while we have the goal of reducing inter-AS traffic."

NIT-PICKY THINGS

Figure 1 is confusing. What is your definition of CDN? The figure makes it look like a CDN is, by definition, a system in which there is no inter-AS bandwidth, but that clearly isn't true. How is a CDN really different from p2p? They're not mutually exclusive.

(*) We agree in the paper that Figure 1 is only a simplification of how a CDN or a p2p system work (now highlighted). We currently state on page 4, section 2.2:

“Swarming p2p systems, and BitTorrent in particular, have gained tremendous popularity recently due to their ability to effectively disseminate content. Figure 1(b)(p2p) shows a simplified illustration of the BitTorrent swarming p2p protocol same as Figure 1(a)(CDN) shows a simplified illustration of CDN operation. The figure is not intended to provide complex details about the protocol, but to convey the key concepts needed to clarify our approach.”

Also, we are aware of the similarities and differences between the two systems and we did not want to burden the paper with definitions. As we already state in the abstract (now highlighted):

“Traditional content distribution networks (CDNs), such as Akamai, replicate content at thousands of servers worldwide in an attempt to bring it closer to end users. Recent years have, however, brought a surge of peer-to-peer (p2p) systems that have demonstrated the ability both to help traditional CDN operations and to effectively disseminate content as independent applications.”

Your opening quotation marks are backwards. Use the back-tick character: ` and ``.

(*) We have changed this.

"one or none leechers" on page 10

(*) We have changed this and we have marked it in the paper.
