A Case for Hybrid Content Distribution for Interactive Video-on-Demand

Andrew MacQuire Andrew Brampton
Nicholas J. P. Race Laurent Mathy
Lancaster University
{*macquire,brampton,race,laurent*}*@comp.lancs.ac.uk*

Michael Fry
University of Sydney
michael.fry@usyd.edu.au

Abstract

With the advent of numerous video distribution services, Content Distribution Networks (CDNs) are under increasing demand. Given the associated expenses, many organisations have made use of Peer-to-Peer (P2P) approaches to offset bandwidth costs. Unfortunately, using clients as part of the delivery process can vastly increase load on service providers' networks. Furthermore, many approaches thus far lack the ability for interactive, on-demand viewing. For these reasons, this paper considers how 'hybrid' methods could alleviate these issues, through simulations of delivery methodologies in a CDN context. As it is unlikely that any one method will suffice for all content, network and audience types, a successful system should adapt in response to changes, as outlined in this paper.

1. Introduction

The rise in the consumption of video content over the Internet has been marked over the past few years. Typically, the means of delivery falls into one of two categories: *push* or *pull*, with both approaches typically involving clients first joining a distribution structure. Push-based systems then send out predetermined segments to clients, regardless of what exactly they may wish to receive. In contrast, pull-based systems rely on responding to requests for segments made by individual clients. The latter therefore typically allows for 'VCR interactivity', in that users may pause and resume playback, and move forwards and backwards within the content.

Push delivery is ideal for any passive broadcast media (*e.g.*, live video). Many clients watching the same content simultaneously means there is a high degree of viewing redundancy that can be exploited. Conversely, direct server-to-client delivery (pull) is commonly used for interactive media, as users are potentially viewing many different sections of content. By adopting a hybrid approach, however, the benefits of both could be leveraged and the disadvantages negated, and an efficient delivery system that exploits redundancy whilst retaining the ability for users to perform VCR operations could be produced.

Through consideration of these approaches, the particular contribution of this work is a study of the effects of user interactivity. Experiments are performed across varied workloads, demonstrating how delivery methods are affected by audience size, content type and the resources available to the system. We simulate large-scale media consumption, using realistically modelled interactions between clients and content sources. Whereas much of the literature in this field focuses on delivering live streams passively, the novelty of this work includes the use of interactivity workloads atypical of previous studies, of the type found in [1], which is both on-demand and interactive.

The remainder of this paper is structured as follows: Section 2 first describes similar work in the field. Section 3 then explains the delivery methods considered, while Section 4 outlines our experimental approach. Section 5 describes the results obtained and their impact, and finally, Section 6 concludes the paper.

2. Related Work

Skevik *et al.* put forth a similar case for hybrid Content Distribution Network (CDN) architectures employing peer-to-peer approaches in [8], albeit with a focus on firewall traversal and avoiding free-riders. Huang *et al.* also focused on the use of P2P with CDNs in [4], noting that a hybrid approach can satisfy the growth of video on the Internet without overloading ISPs. The authors demonstrate this with the use of a technique to chart CDNs in order to localise peer-to-peer traffic. Similarly, Xu *et al.* provided a detailed mathematical and simulated analysis of the issue in [10], looking particularly at peer contribution policies. While we propose a similar approach to these works and others, our analysis is distinct in that we particularly examine the effects of delivering interactive, video-on-demand content, as opposed to the common approach of immutable live streaming.

978-0-7695-3333-9 /08 $25.00 © 2008 IEEE
DOI 10.1109/NGMAST.2008.68

3. Delivery Methods

3.1. Push - Periodic Broadcast over TBCP

The first of our three approaches aims to group similar users together on a multicast substrate. Given the lack of support for network multicast, we use an Application-Level Multicast (ALM) algorithm, specifically the Tree Building Control Protocol [6], which could in practice be implemented by a CDN.

Periodic broadcast can then be used in order to provide the illusion of true Video-on-Demand. Staggered broadcast is its simplest form, involving a number of multicast channels for a given video, and beginning each at an interval evenly distributed throughout its length [3]. Users may then move forwards and backwards throughout the stream by switching channels. The granularity of these operations is, however, limited by the number of channels available. Without allocating a large number of broadcast channels, startup/interaction latency may be unacceptable in longer pieces of media. To allay this issue, enhancements such as harmonic broadcasting can be employed [5]. An adaptation of this method is our 'push' approach to delivering the media, in that content nodes broadcast the same segments of content to as many nodes as possible simultaneously, where TBCP trees are generated as required to act as periodic broadcast channels.

3.2. Pull - Peer-to-Peer

The second major delivery method considered, representative of the 'pull' approach to delivery, is the use of a peer-to-peer, BitTorrent-like protocol. A beneficial side-effect of involving time-sensitive data in such a network is that the incentives that drive the system become more pronounced. In other words, due to the tit-for-tat exchange mechanism, free-riders would be encouraged to contribute, as to improve the probability of receiving the segments they require in a timely fashion. Such protocols are not without drawbacks in the context of streaming media, however.

Firstly, the most common issue is that media playback requires sequential segments of a piece of content; BitTorrent instead delivers them effectively randomly. A solution often proposed to handle the need for sequential delivery is to apply a sliding-window to the data being distributed. Peers then have upper and lower boundaries for the segments which they are interested in downloading, and prioritise their requests accordingly.

Secondly, startup latency should be minimised; BitTorrent's reliance on other peers optimistically unchoking newcomers means this is not the case. Shar *et al.* and Vlavianos *et al.* amongst others have noted and responded to issues such as these [7, 9]. A simple solution is to ensure that established peers perform optimistic unchoking on a more regular basis; Shah and Pâris show that, when combined with a sliding window approach, significant improvements over the base protocol can be achieved in [7]. We therefore employ similar methods to gain the same benefits.

3.3. Hybrid - ALM with P2P Patching

Finally, the third method examined is a 'hybrid' of both the 'push' and 'pull' methodologies. By using TBCP-based periodic broadcast trees in conjunction with peer-to-peer 'patching', video clients can share data enabling them to reach the current broadcast point rapidly. To clarify, when a client joins a periodic broadcast channel, they do not need to wait for the segments to be broadcast to them (*i.e.*, for the channel to reach their desired playback point), but can instead request segments from their neighbours as quickly as possible until they have reached the current position of the multicast channel (a traditional process typically referred to as 'patching').

4. Experimental Setup

Three key variables to be considered for a delivery system are the size of the audience, the nature of the content (workload) and the network resources available. We also vary *peer-to-peer usage*, in terms of the volume of data clients receive from trackers and the number of connections they are allowed to make to other peers. For example, if in an unrestricted environment 100 peers would be returned to an individual client, 10% usage would reduce the number returned to just 10. By varying this value, the effect of peer-to-peer patching on the hybrid method can be observed: at 0% usage, the method is identical to the normal periodic broadcast over ALM, whereas at 100% usage, the peer-to-peer scheme is made use of as much as possible. The increasing effect of the peer-to-peer patching system in the hybrid approach can therefore be observed in an incremental fashion relative to both the pure ALM and pure P2P delivery methods. Naturally, this does not affect the pure ALM approach, as so no variance is seen in its results for plots of this type.

To obtain results that provide a comparison of how particular delivery schemes may handle different pieces of content, varied workloads are considered. These can be classified as follows:

"Start-to-Finish" - Used for baseline comparison, where clients do not use any interactivity features. Media is viewed in a start-to-finish fashion, although the start and end points are not necessarily identical between clients.

"Interactive" - In our study in [1] we produced characterisations and models of workloads derived from real experiences in deploying videos of popular sporting events (the

FIFA World Cup). In essence, the unique aspect of these workloads is large pieces of content with relatively short areas of high interest, highlighting how well a particular approach handles large differences in popularity between content segments.

Beyond the workloads, the metrics used are selected to reflect both network provider and user satisfaction, based on resource usage efficiency and perceived quality of service respectively. The first metric considered, therefore, is **network stress** in terms of the amount of data delivered on the network, normalised to the worst case in a given simulation.

The second metric used is the **average fraction of timely segments received per client** – the average value, on a per-client basis, for the total number of timely segments divided by the total number of segments required by that client. Note that a 'timely' segment is one which arrived before its playback deadline.

Finally, **segment utility** is considered – the number of video segments actually used in playback in comparison with the total number sent over the network. Given that a client may receive segments superfluous to their requirements (*e.g.*, buffering 5 minutes when the client only watches 1), this metric provides a measure of network efficiency relative to the data that clients actually consume.

A high-level overview of the experimental setup is as follows: 1,000 routers exist in a GT-ITM generated topology graph [2]. The graph is of a realistic transit-stub configuration, wherein a single node exists per transit domain, but many exist per stub. Content node(s) are attached to the aforementioned transit node(s), whereas clients are attached to randomly selected members of the stub domain(s).

The group size used in the simulations is 500 clients, and delivery of some subset of the piece of content per client is made using the appropriate method described for each simulation. Each client possesses an individual playback buffer, which should not underflow if user-satisfaction is to be achieved.

In our custom simulation environment, the overall delivery process is modelled in a number of steps over several iterations, whereas the content itself is handled as a series of sequential 'segments'. Firstly, a realistic workload is generated (according to the models from [1]), wherein the clients are provided with individual lists of segments they will be required to obtain within the timeframe of the simulation.

Following workload generation, clients must interact with the delivery structures to begin the acquisition of segments. In the 'pull' peer-to-peer approach, delivery proceeds as described in Section 3.2. The 'push' and 'hybrid' approaches, however, primarily use TBCP periodic broadcast trees, meaning the initial step is for clients to determine which tree(s) will contain the segment(s) they require, and where they are rooted (*i.e.*, the address to join). In the

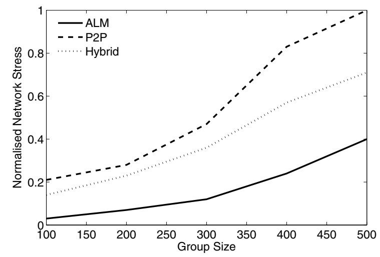

Figure 1. *Network cost of delivering an interactive workload, with varied group size*

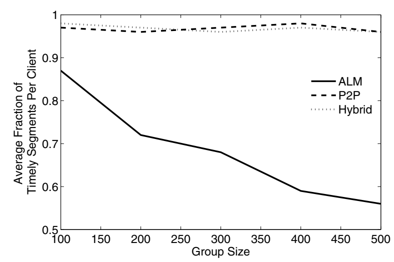

Figure 2. *User satisfaction during delivery of an interactive workload, with varied group size*

simulations this is achieved through a lookup process on a single node for simplicity, although in a real-world deployment with numerous videos of many segments each, a more sophisticated arrangement may be appropriate.

Following location of the correct tree for the required segment(s), a client must then wait on the periodic broadcast to roll-around to the required playback point for consumption. In the hybrid approach, the peer-to-peer mechanism can now be used to speed this process somewhat, by attempting to rapidly 'patch' required segments that the client would otherwise have to wait for. Note that clients also have to use the knowledge of segment-tree mappings obtained from the lookup service to anticipate when trees must be switched, as to avoid suboptimal resource usage through nodes being present in a tree unnecessarily.

5. Experimental Results

Figure 1 shows the resulting network cost across the delivery methods for an interactive workload with varying group sizes. The amount of traffic created for the ALM-based methods is shown to be lower than pure peer-to-peer; significantly so for large groups. A delivery method being low cost, however, is of little use unless it can deliver an equivalent quality of service. Figure 2 shows that this is not the case for pure ALM, as an initially high level of timely

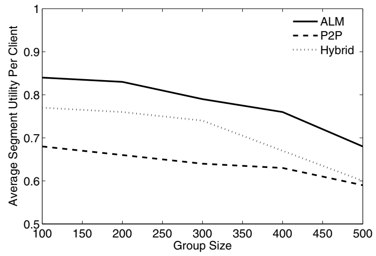

Figure 3. *Overall segment utility during delivery of an interactive workload, with varied group size*

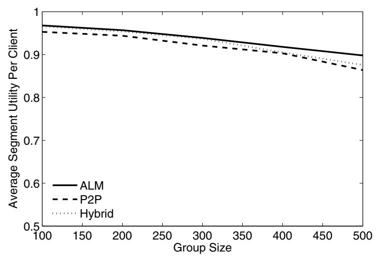

Figure 4. *Overall segment utility during delivery of a start-to-finish workload, with varied group size*

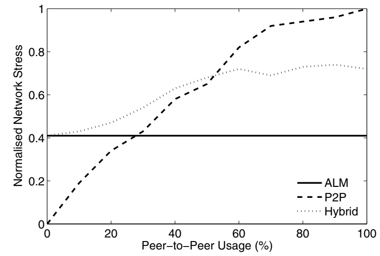

Figure 5. *Network cost of delivering an interactive workload, with varied peer-to-peer usage*

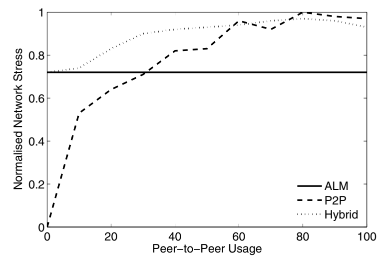

Figure 6. *Network cost of delivering a start-to-finish workload, with varied peer-to-peer usage*

segments reduces rapidly with increasing group size. Such a result may be indicative of this approach being unable to handle large, interactive groups well, possibly due to the joining overhead associated with switching between highly populated broadcast trees. Indeed, when delivering a 'start-to-finish' workload (not shown due to space restrictions), pure ALM was found to provide a level of performance similar to the other methods for the considered metrics.

Figures 3 & 4 show the average fraction of useful segments that were delivered during each simulation on a per client basis. Interestingly, Figure 3 indicates that for a high-interactivity workload, up to half the segments on the network could be sent fruitlessly, with the factor of interactivity separating the delivery methods noticeably. In contrast, Figure 4 shows that when the workload is start-to-finish, each of the schemes achieves a similar level of overall utility, working at an efficiency of around 85% upwards at all times, with little to separate the methodologies. The 'wasted' segments in these cases are most likely due to congestion, and accordingly show an increase with group size.

5.1. Varying Peer-to-Peer Usage

Figure 5 shows the network stress in terms of data delivered per link for an interactive workload, normalised to the 'worst case' of full peer-to-peer usage. As simulations run to the point where clients no longer have any interest in receiving segments, the reduced traffic in the low usage scenarios for the peer-to-peer case is to be expected. Two notable points on this plot are around 25% and 50% peer-to-peer usage, where the P2P method crosses over the ALM and hybrid approaches respectively. Following the latter intersection, the hybrid method appears to level off while pure P2P continues to increase. Dependent on the user satisfaction for these schemes at these points, this indicates that the hybrid method can produce a consistently lower amount of network traffic relative to pure P2P, although the addition of patching is resulting in twice the amount produced by ALM

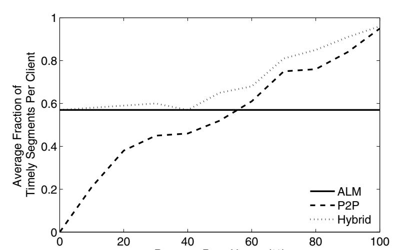

Figure 7. *User satisfaction during delivery of an interactive workload, with varied peer-to-peer usage*

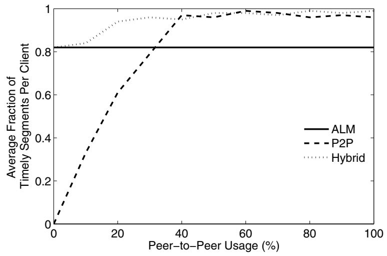

Figure 8. *User satisfaction during delivery of a start-to-finish workload, with varied peer-to-peer usage*

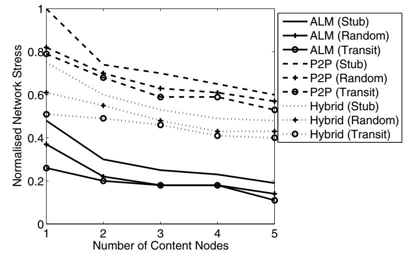

Figure 9. *Effect of additional shared-content nodes on overall network stress*

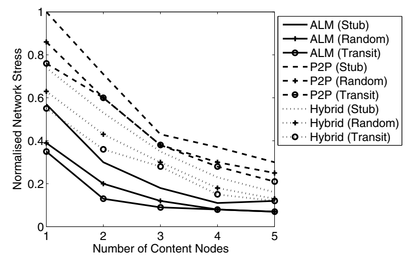

Figure 10. *Effect of additional redundant-content nodes on overall network stress*

alone. A key point to consider regarding the high network cost of P2P is that the ALM-based approach is exploiting the concept of locality in building shortest-path trees to the highly provisioned content nodes. In the peer-to-peer case, this does not happen, and many segments are likely being sent lengthy distances between groups of clients viewing the same section of video, potentially across multiple network transit/stub domains.

If the first figures are considered as showing network cost, then the likes of Figure 7 show the resultant level of 'user satisfaction', in terms of the average fraction of timely segments per client. Recall that two particular points of interest in Figure 5 were around the 25% and 50% peer-to-peer usage marks. In this figure, however, no significant improvement over pure ALM is shown until around the latter of these points: the 50-60% mark. It therefore seems that without the ability to disseminate a large amount of P2P data, the use of the P2P approach just results in additional overhead for this type of workload. As the usage increases beyond this point, however, the number of timely segments being delivered increases significantly for both the schemes that make use of P2P. In almost all cases, however, the hybrid method outperforms pure peer-to-peer, although not by a large amount. This may be due to the sequential nature of the media playback – when periodic broadcast is used, most of the data that clients need will be pushed to them anyway – the pull approach may only be particularly useful when clients have to seek to new points in the media. The high level of interactivity in the workload used for this particular figure may therefore explain the relatively poor performance of pure ALM.

Figure 6 shows the result of a simulation similar to that conducted for Figure 5, but with users consuming the content 'start-to-finish'. The network cost is found to be closer between the three methods in this case than with a high-interactivity workload, supporting the idea that user interactivity can be more costly for certain delivery methods. Interestingly, when examined in conjunction with Figure 7, it can be observed that the pure peer-to-peer approach outperforms the hybrid method under these metrics when "peer-to-peer usage" is around the 40% mark. This particular simulation result may therefore run contrary to the initial expectation that periodic broadcast may be better suited than pure peer-to-peer for many users viewing the media in a generally sequential fashion. Also note, however, that the average fraction of timely segments for pure periodic broadcast in this low-interactivity scenario is significantly better; effectively double that of the high-interactivity workload, and more akin to the results obtained for the high-interactivity workload with small group sizes. This result is as expected, as reduced interactivity correspondingly results in fewer clients jumping between broadcast trees, fewer setup delays, and thus fewer untimely segments according to the clients' demands.

5.2. Additional Nodes

In all cases, as shown in Figures 9 & 10, the addition of extra resources in the form of additional content sources (*i.e.*, the roots of trees or seeds in swarms) is beneficial, with similar trends observed regardless of the delivery method being used. This is especially true of the case when content is made available in a redundant fashion rather than simply being 'striped' across nodes. The exact placement of these nodes also has some impact, albeit a less significant one, with a slightly diminishing effect as more nodes are made available.

5.3. Key Observations

From these findings, several points become clear. Regarding network costs in terms of traffic generated, peer-to-peer is relatively expensive in comparison with the simpler ALM approach. This cost is offset, however, by a resilience to large group sizes and high levels of interactivity in comparison with ALM. Fortunately, when the methods are combined into the hybrid method, network cost is lower than the peer-to-peer case, albeit higher than ALM alone, making it an effective compromise. In all cases, larger group sizes result in an increase in the amount of network traffic, and this also has a slightly negative effect on segment utility.

In terms of user satisfaction, larger groups have little effect on the fraction of timely segments received when the peer-to-peer system is involved under any level of interactivity. This is also largely true for the case of pure ALM when low levels of user interactivity are observed, but when users begin to jump around within a video, ALM performs significantly worse. This effect can be attributed to the lengthy join delays when the broadcast trees are particularly long, coupled with the standard waiting period on the broadcast channel to receive the segment(s) the user requires. Naturally, this does not affect the 'pull' approach of the peer-to-peer method, and when this is combined with the ALM scheme, the peer-to-peer system acts as a means of 'patching'; providing the video segments required in an on-demand fashion until the 'push' scheme has stabilised. For similar reasons, segment utility is quite similar across all methods for low interactivity workloads, but when high levels of interactivity occur, there is a marked difference between the schemes. For instance, while the peer-to-peer approach typically achieves high-levels of user satisfaction, a larger percentage of segments are wasted relative to ALM. The naturally sequential nature of the broadcast over ALM may therefore be acting in its favour here, given that users are highly likely to want large numbers of video segments delivered in this fashion.

6. Conclusion

From the results observed, we can conclude that while peer-to-peer methodologies are certainly feasible for the delivery of interactive content from the user perspective, they are somewhat network-intensive. In contrast, a more traditional approach based around classical periodic broadcast techniques over an application-level multicast structure apparently work well for smaller numbers of passive viewers, but encounters problems when user interactivity and group size increase. The combination of these two approaches, broadly 'pull' and 'push' respectively, can, however, offer a good compromise that provides adaptability to varying conditions in terms of audience size, interactivity levels, and the resources available. Across all the delivery methods considered, providing additional resources in terms of extra content nodes is beneficial, with their placement relative to clients being increasingly important dependent on their abundance. It is therefore apparent that mixing a live distribution approach such as application-level multicast with an appropriate peer-to-peer 'patching' mechanism over a typical network infrastructure (*i.e.*, typically lacking in multicast support) can provide a workable solution for delivery of on-demand video with interactivity support in a CDN environment. Given the wide variety of possible workloads, delivery methods and variables, much potential exists for future work in this field.

References

[1] A. Brampton, A. MacQuire, I. A. Rai, N. J. P. Race, L. Mathy, and M. Fry. Characterising user interactivity for sports video-on-demand. In *Proc. of NOSSDAV'07*, July 2007.

[2] K. Calvert and E. Zegura. Georgiatech internetwork topology models. http://www.cc.gatech.edu/projects/gtitm.

[3] A. Dau, P. Shahabuddin, D. Sitaram, and D. Towsley. Channel allocation under batching and VCR control in video-on-demand systems. *Journal of Parallel and Distributed Computing*, 30(2):168–179, November 1995.

[4] C. Huang, A. Wang, J. Li, and K. W. Ross. Understanding hybrid CDN–P2P: Why Limelight needs its own Red Swoosh. In *Proc. of NOSSDAV'08*, May 2008.

[5] L.-S. Juhn and L.-M. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, September 1997.

[6] L. Mathy, R. Canonico, and D. Hutchison. An overlay tree building control protocol. In *Proc. of the 3rd International Workshop on Networked Group Communication (NGC)*, November 2001.

[7] P. Shah and J.-F. Pâris. Peer-to-peer multimedia streaming using BitTorrent. In *Proc. of the 26th International Performance of Computers and Communication Conference (IPCCC 2007)*, April 2007.

[8] K.-A. Skevik, V. Goebel, and T. Plagemann. Design of a hybrid CDN. *Lecture Notes in Computer Science*, 3311(2004):206–217, 2004.

[9] A. Vlavianos, M. Iliofotou, and M. Faloutsos. BiToS; enhancing BitTorrent for supporting streaming applications. In *Proc. of the 9th IEEE Global Internet Symposium (GI2006)*, April 2006.

[10] D. Xu, S. S. Kulkarni, C. Rosenberg, and H.-K. Chai. Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems*, 11(4):383–399, 2006.