Virtual Multicasting as an Example of Information Mass Transit

Philip Machanick[a] Brynn Andrew[b]

Department of Computer Science, University of the Witwatersrand,
[a]*philip@cs.wits.ac.za*, [b]*bandrew@cs.wits.ac.za*

Abstract

Virtual Multicasting (VMC) is a specific instance of a more general idea, Information Mass Transit (IMT). IMT aims to reduce the waste of bandwidth resulting from individual streams of data, while improving user-level latency. By analogy with mass transit where shared transport reduces the load on infrastructure, IMT aims to use networks and other infrastructure more efficiently. VMC combines some of the benefits of caching (transparency, dynamic adaptation to workload) and multicast (reducing duplicated traffic).
Keywords: *improved bandwidth utlilization, content delivery, multicast, internet caching*
Computing Review Categories: *C.2.6, C.2.2*

1 Introduction

Information Mass Transit (IMT) is a general design philosophy aimed at exploiting commonality of data on a medium to reduce bandwidth demands and improve latency [8]. The name derives from an analogy with mass transit, where apparently-slower modes of transport like buses and large passenger aircraft are faster for moving large numbers of people with common destinations than apparently faster alternatives (cars, executive jets). Sharing a common form of transport reduces congestion, and makes better use of common media (in the transport case, roads and airports).

Internet congestion is a growing problem: as capacity increases, so does demand. Given that there could be significant common traffic at peak times, it seems reasonable to investigate sharing common data as far as possible. By analogy with the mass transit idea for moving people, if much traffic at the same time is similar, grouping this similar traffic could have significant advantages.

Virtual Multicasting (VMC), as a specific instance of IMT, finds common streams which may have started at similar times, and combines them. This general model can differ considerably in different implementations. For example, grouping FTP streams may not introduce significant latency or real-time concerns, provided the streams are sufficiently large that saving in transmission time dominates any cost of additional latency.

In the case of playing a movie, latency may be a concern, if VMC results in a perceptible delay to an existing viewer. However, by contrast, if many viewers are watching the same real-time video or audio stream, there is some tolerance of lost traffic and if sharing the stream significantly reduces bandwidth requirements over having individual streams, VMC will be a win in terms of perceived quality of service.

1.1 Information Mass Transit

A number of applications of the IMT idea have been proposed [8].

The general model is one of sharing a stream for multiple purposes; the actual realization may differ considerably in specific cases.

One example is the Scalable Architecture for Video on Demand (SAVoD), which aims to implement a video on demand system which scales up to an unlimited number of users [7].

The SAVoD architecture works by streaming multiple instances of a movie continuously, so that a virtual VCR can be implemented by finding a suitable point in any given stream, to perform operations such as fast forward, rewind, or start a new movie. The principle is to invest in a large amount of bandwidth, with the goal of removing all requests to the server. Consequently, the biggest latency problems in scaling up to unlimited users are removed.

The VMC idea is the next attempt at realizing the broader IMT idea.

1.2 Virtual Multicasting

Virtual Multicasting (VMC) is an attempt at exploiting short-term latencies in Internet traffic, particularly higher up the bandwidth hierarchy. A high volume of similar traffic may periodically occur as new software is downloaded, a large number of clients join the same audio or video stream, or visit a new web site.

Such traffic cannot easily be cached for two reasons:

- the repeated traffic may be transient, and the demand may no longer exist by the time it is cached
- the users may be widely spaced around the Internet, and only the higher-bandwidth links at the top of the hierarchy may see duplicated traffic, i.e., endpoints are not the right place to cache this kind of traffic

The transient nature of this kind of similar traffic also makes multicast an inadequate solution to the problem of reducing wastage of bandwidth.

This paper presents some preliminary data on VMC, and proposes further research.

1.3 Remainder of Paper

The remainder of this paper is structured as follows.

Section 2 provides an overview of the VMC concept and related approaches, and relates it to the more general IMT model. Section 3 outlines some preliminary results to support the concept. Proposed work is described in Section 4. Finally, conclusions are presented in Section 5.

2 Virtual Multicasting

Virtual Multicasting is an attempt to avoid or control congestion on the Internet. It does this by moving away from the traditional model of content delivery (unicast) to one that makes more effective use of the available bandwidth. Instead of having data distributed from a single point, VMC aims to distribute the dissemination of data, reducing the congestion of servers and interconnected networks, freeing bandwidth and as a result, reducing latency from a user's point of view.

Virtual Multicasting is intended to be implemented as an extension of IP routing, in which common TCP streams are identified, and combined. As opposed to standard multicasting [4], there is no explicit setup, and if a client joins a stream late, it will receive earlier traffic out of sequence, sent as a separate stream.

VMC works by maintaining a record of data travelling on the router. if a new client requests data that the VMC router is transmitting already, the request is not passed to the server. Instead, the router creates a request for the previously-transmitted portion of the data, and copies the current stream to the new client.

The router now has two clients receiving the same data from a single source.

Once the download is complete for the first client, the clients which joined the VMC session later issue a request for data they missed.

VMC can be contrasted not only with multicasting, but also with proxy caches, which save recent content to avoid repeated delivery. VMC differs from caching in that it occurs in the highest-traffic segments and routers, rather than at the endpoints. Further, VMC happens on the fly, whereas caching stores a stream for future use. VMC therefore exploits very short-term locality, and locality across a different part of the Internet.

Ideally, VMC should be completely transparent. However, in our initial work, we are prepared to make simple modifications to standard protocols to demonstrate feasibility.

The remainder of this section provides a brief overview of conventional multicasting, proxy caches and an experimental VMC implementation.

2.1 Multicasting

IP multicasting is the transmission of a packet to a subset of hosts in a network [5]. It provides packet delivery to these hosts at a lower network and host cost than broadcasting to all hosts or unicasting to each host in the group. It offers efficient multi-destination delivery and robust unknown destination delivery to hosts on a network [5].

One of the major problems with enabling multicasting throughout the Internet is the lack of standardisation for multicast protocol and specifically the implementation of the protocols. As a result, multicast groups are mismanaged, meaning that data cannot be distributed to hosts.

Another problem is that many routers on the Internet are not configured to allow the transmission of multicast packets. These routers have to be bypassed by IP tunneling [11], a non-trivial task and as a result multicasting is not widely supported by Internet Service Providers (ISPs).

Finally, the "best-effort" attempt at data delivery that multicast operates with, is not good enough for many applications which need data to be reliably transferred. Reliable multicast protocols have been developed, but they are inefficient in the delivery of data and have a propensity to cause packet storms [6].

2.2 Proxy Caching

A proxy cache (often simply referred to as a "cache") is an application that is installed between Web servers and clients. It watches requests for Web objects (HTML pages, images and files) and saves a copy of the object locally. Subsequent requests for the same object can then be served from the cache.

Caches can reduce latency as seen by clients and reduce the bandwidth used by the clients behind the cache. Caches can be seen as congestion control mechanism, since they reduce traffic on the Internet by storing data locally.

Some incoming data cannot be cached. This is due to factors such as dynamic content and rapidly changing web pages. Studies have shown that the amount of Web traffic that cannot be cached is as high as 20% [10]. Furthermore, even with an infinite cache size, the upper bound for the hit rate is 30-50% [1, 10].

It is not always useful to have a cache hit, because the cache server may be overloaded and unable to serve the object efficiently [9]. Furthermore, the time taken to check the validity of the object might be longer than retrieving the object itself. Caches may also be slower on misses than an uncached connection, since the time taken searching a hierarchy for the object may be longer than retrieving the data from the origin server [10]. Every slowdown in the cache adds to the latency experienced by the user.

Finally, caches are often large, expensive pieces of hardware and software that have to be configured and constantly maintained. If there is a problem with the cache server, an entire network may be deprived of Internet connectivity, which may be unacceptable for many applications (e.g. Internet banking).

2.3 Comparison to VMC

The common basis of multicasting and caching is that they are bandwidth saving and congestion reduction mechanisms. VMC uses the single data stream idea of multicasting and the transparent nature of caching to produce a mechanism with the benefits of both, without the costs and problems of multicasting and caching.

Unlike caching, VMC occurs near the top of the hierarchy, so the cost is only incurred at high-throughput routers, whereas caching occurs at endpoints, and is therefore a highly replicated cost. Multicasting requires prior knowledge that a stream will be shared, and has a high setup cost. VMC, by focusing on traffic through the highest-traffic routers, reduces the setup cost. Further, the VMC approach of transparently initiating sharing when it is detected means that it is not necessary to predict the need for sharing in advance.

2.4 Experimental VMC Implementation

Establishing the feasibility of the VMC approach takes a number of forms. First, the actual mechanics of VMC have to be developed and demonstrated. Second, it will be no good if the method exists in a vacuum, so good interaction with the current Internet protocols must be demonstrated. Finally, VMC is likely to add latency. This additional latency must be measured and weighed against latency gains, to decide the effectiveness of the method.

In order to evaluate these feasibility issues, an experimental VMC system will be built. The development of this system will also mean that the extent of interaction with current Internet protocols can be measured.

Thereafter, experimentation will be done to measure the overhead incurred by the method and quantify the benefits that the method could bring in terms of bandwidth and latency savings.

The Virtual Multicasting router software will be implemented using the Netfilter [3] filtering mechanism.

3 Preliminary Experimentation

A preliminary study of FTP logs from a commercial Internet service provider showed that there is significant overlap of FTP traffic, at least from their site. The overlap of traffic would not occur with VMC, since streams would be sharing this data.

By doing very rudimentary calculations (not taking bandwidth and latency issues into account), the amount of data shipped by VMC can be compared to data shipped normally. The calculations were done over 11 consecutive days of logged traffic.

The comparison of the number of bytes shipped can be seen in Figure 1. The x-axis represents the different days on which the logs were taken.

Figure 1 indicates that the number of bytes shipped using Virtual Multicasting is less than normal data transfer. A cumulative comparison is in Figure 2. Again, the number of bytes are plotted per log-day.

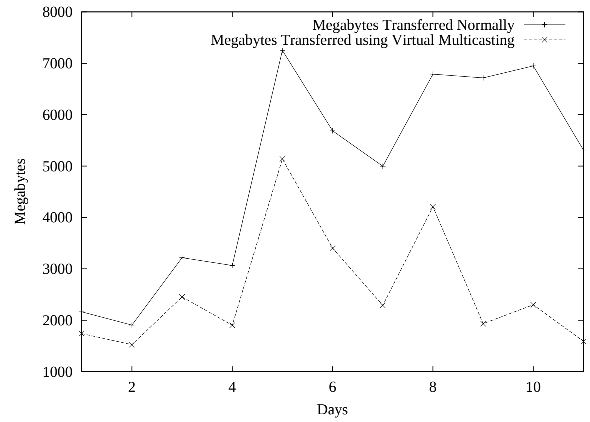

Figure 1: *Comparison of Transfer Methods*

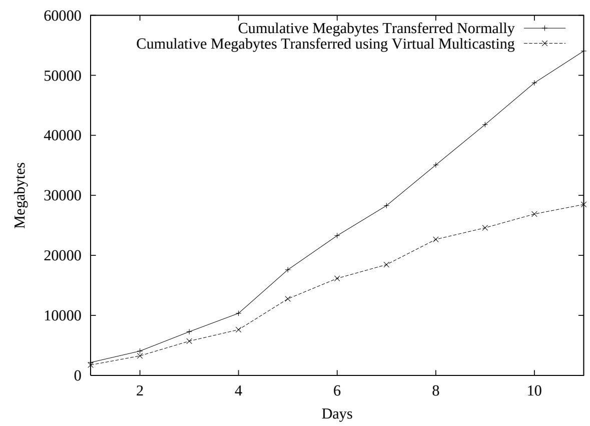

Figure 2: *Cumulative Comparison of Transfer Methods*

Figure 2 further demonstrates the potential for better performance of Virtual Multicasting.

The total number of bytes transferred normally over the log days examined was 5.67×10^{10}. The number of bytes transferred using the Virtual Multicasting approach was 2.99×10^{10}, 52% less than the normal mode of transfer. The biggest saving through the implementation of Virtual Multicasting was 71% and the smallest was 19%.

This initial study shows that VMC has considerable promise, and is worth further investigation.

4 Proposed Work

The main goal of this research is to provide a feasibility study of VMC. It is thus necessary to focus on the problems with VMC and its implementation rather than the details of the protocols we are working with.

While FTP shows considerable promise, the FTP protocol does not lend itself to simple modification to test the ideas further. We intend therefore to follow up with a more detailed investigation of protocol changes to HTTP to enable VMC [2].

HTTP encapsulates all the file transfer mechanisms of

FTP and is widely used as such. Furthermore, the protocol itself is cleaner and better defined – particularly for our purposes.

Once feasibility is demonstrated, other protocols will be investigated. Ideally, VMC should be transparent, but the initial focus is on demonstrating the benefits, rather than on the most convenient implementation.

5 Conclusions

This section summarizes our preliminary results, and proposes further work. Finally, we conclude by considering the overall potential of both VMC and IMT.

The preliminary results show the potential for a significant saving using VMC. Considering that this study did not take network load into account, even better results can be expected when the mechanism is accurately modeled and simulated. More importantly though, the results show that Virtual Multicasting is an idea worth further development.

Further work on IMT includes investigation of implementation issues for SAVoD, and investigation of further application of the principles in other areas.

We further propose to investigate areas where VMC can be implemented transparently, and modifications to standard protocols where it cannot be implemented transparently.

VMC is a promising approach to explore further. Once we have completed our initial implementation, we will aim to report results such as the trade-off between extra costs of VMC and the benefits. We expect that the overall benefit will outweigh any overheads we introduce, but measurement will be required.

In general, the IMT approach is promising. Internet bandwidth scales with users as well as with new technology, and general experience has been that traditional models of communication very quickly result in loss of the benefit of new bandwidth.

We believe that a new approach is called for, and IMT (including its particular manifestations, SAVoD and VMC) attempts to address this need.

References

[1] Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, Stephen Williams, and Edward A. Fox. *Caching Proxies: Limitations and Potentials*. WWW Document, December 1995. `http://ei.cs.vt.edu/~succeed/WWW4/WWW4.html`.

[2] B Andrew and P. Machanick. The virtual multicasting approach to bandwidth conservation. In *Proceedings of SATNAC 2000*, Somerset West, South Africa, September 2000. In press.

[3] Marc Boucher and Rusty Russell. The Netfilter project: Packet mangling for Linux 2.3+. WWW Document, 2000. `http://netfilter.kernelnotes.org`.

[4] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153–162, April 1996.

[5] Stephen E. Deering and David R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, February 1990.

[6] Brian Neil Levine. *A Comparison of Known Classes of Reliable Multicast Protocols*. Master's thesis, University of California, Santa Cruz, 1996.

[7] P. Machanick. Design of a scalable video on demand architecture. In *Proceedings of SAICSIT '98*, pages 211–217, Gordon's Bay, South Africa, November 1998.

[8] P Machanick. Streaming *vs*. latency in information mass-transit. *Computer Architecture News*, 26(5):4–6, December 1998.

[9] Harrick M. Vin Renu Tewari, Michael Dahlin and Jonathon S. Kay. *Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet*. Technical Report TR98-04, The University of Texas at Austin, 1998.

[10] Alex Rousskov and Valery Solokiev. On Performance of Caching Proxies. WWW Document, August 1998. `http://www.cs.ndsu.nodak.edu/~rousskov/research/cache/squid/profiling/papers`.

[11] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, third edition, 1996.

Received: 7/00, Accepted: 8/00

Acknowledgments

We would like to thank The Internet Solution for providing logs on which our preliminary study is based. This work has been supported by the National Research Foundation.