A Novel Approach for Transparent Bandwidth Conservation

David Salyers, Aaron Striegel*

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN. 46556 USA
dsalyers@nd.edu, striegel@cse.nd.edu

Abstract. In this paper, we present a novel approach, stealth multicast, which offers a practical solution for the adoption of network-level IP multicast. Rather than focusing on a global scale such as with previous approaches, stealth multicast optimizes efficiency on a domain-wise scale. In short, stealth multicast dynamically combines redundant data payloads into virtual groups for multicast transmission across the domain. At the edge of the domain, the packets are converted back to unicast, thus keeping stealth multicast true to its namesake in that neither the user applications nor the external domain are aware of the presence of multicast. We present simulation studies to show the potential of stealth multicast.

1 Introduction

As the Internet has grown and evolved, the demands on the network have shifted from simple connectivity to more sophisticated demands. The point-to-point nature of the Internet has created an increasing presence of redundant data as the applications using the network increase in both scope and scale [1, 2].

A wide variety of techniques have emerged to increase the efficiency of the network. They can be generally classified into two categories: multicast (active) and caching (passive). The first category, IP multicast [3] and application-level multicast (ALM) [4], require that the application and network work together in order to provide maximum efficiency. These technologies are generally hampered by their requirement for global deployment. Although ALM does not have the requirement for network modification, it still requires modifications to the application and can add substantial delay to the transmission.

The second category, is cache based approaches (packet caching [2] and media caching [5, 6]). Although cache-bases approaches do not require the global support of the network or the application, cache based approaches only work for long term redundancy rather than short term (as in multicast).

The contrast of multicast and caching-based approaches to bandwidth conservation introduces the motivation for this paper. We propose a new protocol

---

for bandwidth conservation, *stealth multicast*, which provides the deployment simplicity of caching while handling data with short term redundancy normally associated with multicast. Unlike traditional approaches to multicast (IP multicast or ALM) that can require cooperation among various parties using the service (i.e. application support, inter-domain routing), stealth multicast conceals the multicast transport in a domain (or beyond) by dynamically converting packets to and from multicast at the edge of the domain. Hence, stealth multicast frees an ISP from relying on any external participation to yield an immediate, tangible benefit to network performance.
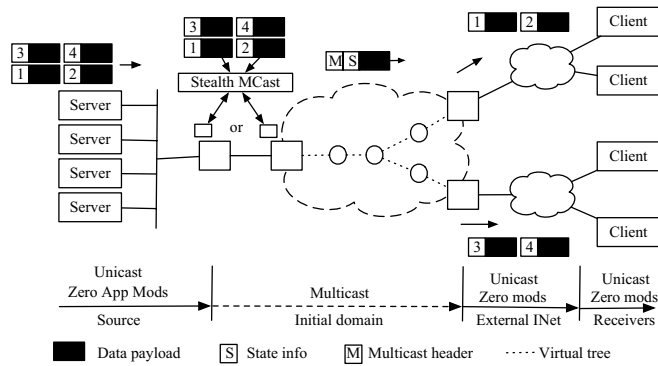

Fig. 1. Stealth Multicast - Overview

2 Stealth Multicast Overview

Figure 1 illustrates the proposed model and its fundamental concepts wherein a server dispatches information to four separate clients using separate unicasts. The key component of the model is the stealth multicast module which assembles candidate packets into *virtual groups* for multicast transmission across the network. The virtual groups themselves are constructed dynamically[1], based on redundant data payloads from the same source application, and are assembled at the *Virtual Group Detection Manager* (VGDM). Packets that are amenable to multicast (which is determined by the background traffic analysis engine) are queued into multicast groups, and packets that will not benefit from multicast are immediately forwarded without incurring any extra queuing delay. The notion of stealth comes from the fact that the entire process itself is hidden and nearly unnoticeable to the external Internet. The sequence of events is described in more detail below.

---

[1] The virtual group itself may have little or no relation to the physical group (actual multicast tree).

First, the application transmits packets with the same data payload to different clients. A digital signature is then created for the data payload (at the VGDM) that uniquely identifies the data payload [2]. The main attributes of the packet (packet size, signature, source IP, and source port) are passed to a background traffic analysis engine. This analysis engine is used to determine if a packet is likely to benefit from multicast. If not, the packet is not queued and is immediately forwarded. If the packet is to be queued, it is queued into a virtual group that shares the same attributes as itself or a new virtual group is created. The packet is then released for transport when a predefined condition is met (group size, timer, etc.). If the redundancy realized is sufficient, the virtual group is transmitted via multicast(PIM-SSM [7]) after selecting the appropriate multicast tree. If not, the packets are forwarded using standard unicast transmission. The unique portions of each packet are preserved as state information. Finally, the packet arrives at the egress point for the domain where the packet is converted back to the original unicast packets using the state information for the virtual group. To the external domain, the unicast packet is indistinguishable from the packet that arrived at the VGDM.

Critical to the ISP, stealth multicast is a sender-driven approach to multicast rather than the receiver-driven approach of network-level IP multicast. The entire multicast process (edge router join/leave) is controlled by the VGDM. Thus, an ISP can easily assess the resource cost and benefit associated with each stealth multicast transmission. Furthermore, stealth multicast is a domain-wise solution rather than an end-to-end solution. Thus, stealth multicast is interested only edge-to-edge transport rather than in end-to-end transport, which improves deployability.

Key Principles: Stealth multicast adheres to two key principles, namely *external transparency* and *QoS impact*. The first principle, external transparency, focuses on deployment. Stealth multicast is application agnostic, and hence, global inter-operability issues and other complexities associated with network-level multicast operation are removed.

The second principle ties into the first principle and into the stealth of the model itself. A significant QoS change may impact the functionality of the applications utilizing the network. Although a positive QoS impact may not necessarily generate criticism, a negative impact on QoS will certainly cause issues with application functionality.

Additional Benefits: First, and most importantly, stealth multicast removes the hurdle of application development to reap a benefit from multicast. Unlike ALM which requires changes to the server and clients, stealth multicast operates transparently to the applications, thus co-existing seamlessly rather than forcing changes to the application. While this does introduce a tradeoff with regards to efficiency, the removal of a dependency on application development increases the immediate appeal of stealth multicast.

Second, the transparency of stealth multicast allows it to be one of the first models to directly address the economic incentives for ISP adoption of multicast. Unlike existing approaches to multicast where the incentive is almost entirely

for the user, the economic benefit of stealth multicast is directable. By varying the location of the VGDM (after the uplink, before the uplink, etc.), an ISP can directly control the benefit of multicast. Furthermore, the sender-driven nature of stealth multicast makes the cost of multicast distribution immediately known at the ingress, thus simplifying shaping as well as offering the opportunity for efficiency-based pricing.
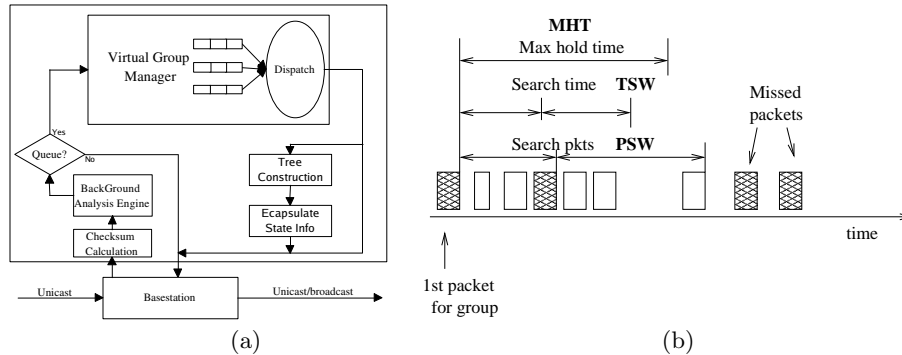

Fig. 2. (a) VGDM - Basic Components (b) Virtual group search settings

3 Stealth Multicast Operation

For conceptual purposes, the VGDM can be viewed as a collection of COTS hardware dedicated to serving an uplink whose traffic can benefit significantly from stealth multicast. Figure 2(a) shows the components involved once the packet arrives at the VGDM which are discussed below.

Signature Generation: Once a packet arrives at the VGDM, it is uniquely categorized according to its data contents. This is done by generating a signature for the payload of the data packet (this does not include the header information of the packet). The signature is computed using an MD5 checksum on COTS hardware [2][2], which will give a checksum that is sufficient in that no two unique packets of the same size will calculate to the same signature.

Background Traffic Analysis Engine: Once the signature has been calculated, the pertinent packet information (source IP, port, data size, and signature) is sent to the background traffic analysis engine. The background analysis engine is used to track if a specific source IP and port is likely to produce packets that will benefit from stealth multicast. The potential for benefit is determined

---

[2] Shown to be easily done on a Pentium II 300Mhz machine running Linux 2.0.31 at speeds greater than 45Mb/s when running in user-space.

by keeping track of if the data from a specific source IP and port would have been mulitcasted had it been queued. The engine is then queried to determine if the packet should be queued for stealth multicast or be immediately forwarded because it is not likely to benefit from stealth multicast.

When the analysis engine is queried, there are three possible cases. The first is that the combination of source IP and port have not been seen before. If this is the case, a new entry is created in the analysis engine and the packet is forwarded along under normal unicast transmission. The next possible outcome is that the source IP and port are known not to benefit from stealth multicast; again in this case the packet is immediately forwarded using unicast. The final possible outcome is the source IP and port are known to be amenable to multicast.

3.1 Virtual Group Management

If the packet is known to be a good candidate for stealth multicast, it is passed to the virtual group manager for placement into virtual groups. Three different trigger mechanisms are used to define the performance (additional multicast efficiency) as well as the penalty (queuing delay) introduced by stealth multicast. The goal is to balance these two trade-offs.

The three dispatch triggers (shown in Figure 2(b)) are **MHT**, **TSW**, and **PSW**. MHT is the maximum amount of time that a virtual group can be queued, it places are hard upper bound on the impact queuing will have. TSW is the inter-packet delay between two packets in a virtual group. Finally, PSW, is a hard limit to the number of packets to scan before a group is released. Once a packet is triggered for release, it is given to the transport mechanism for dispatch via multicast or unicast (insufficient virtual group membership). A more detailed description of the triggers for virtual group release is presented in [8].

Overflow: The memory requirements for the VGDM are quite low. However, in the case of overflow the packets are simply not queued and forwarded as unicast transmissions to their destinations. A more thourough discussion can be found in [8].

3.2 Multicast Transport

If the packet is to be multicast the next dominant issue is how to transport the packet across the domain. While IP multicast operates in a receiver-driven approach, the actual makeup of the receivers in stealth multicast is not known *a priori* and may be highly dynamic depending upon the accuracy of the virtual group detection mechanism. However, this problem is somewhat simplified as multicast transport need only be concerned with transport across the domain whereby the packet is converted back to unicast. Thus, a receiver in the context of a virtual group is the egress point for the domain where conversion occurs rather than the end point (client).

For transport, we propose to employ a dynamic approach whereby multicast groups are created/updated to satisfy all potential egress points for a given

source application. In essence, multicast groups are created on a 1:1 ratio with the amenable sources that the VGDM has detected such that each multicast group is a superset of all likely egress points for the source application (see Figure 3). Unlike the broadcast approach whereby all egress points are on the tree, and hence significant bandwidth may be wasted, the egress points are added to the group only as necessary. In the event that an egress point is not yet in the multicast group for the source application, the packet can simply be sent onwards via unicast in the interim.

To avoid any major deployment issues, the dynamic trees are created by coupling extra control messages with PIM-SSM functionality. For all trees originating from stealth multicast, the VGDM is considered the source since it is the initial point for conversion to multicast. We assume that the VGDM knows the identity of all of the egress points (either by configuration or discovery[3]), and that a broadcast tree containing all of the egress points has been created. In order to drive group-wise changes, the VGDM broadcasts control messages containing a list of PIM-SSM commands for the egress nodes to execute. The commands are included via XML to allow for simple inter-operability between a VGDM and an edge router (see [8]). With this capability, the receiver-driven nature of multicast is offered as a sender-driven interface for the VGDM. In addition, the control messages may be augmented to request acknowledgment, QoS-based reservations, and other third party extensions due to the XML basis of the messages.
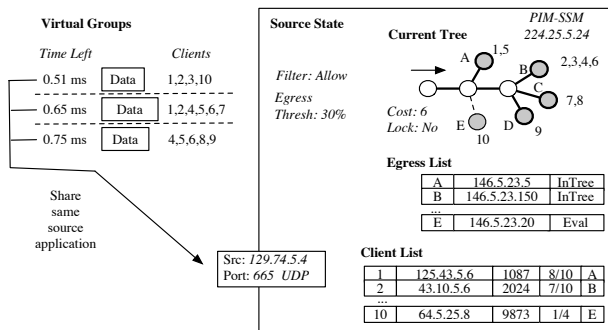
Egress Point Selection: The notion of dynamically created trees offers several unique design challenges for egress point selection that must be addressed.

- *Virtual Join:* What is the threshold for an egress point to officially become part of the tree? If the VGDM sees only one packet going out a specific client (and its respective egress point) and no future packets, it does not make sense to automatically add the egress point.
- *Virtual Leave:* When does an egress point get purged from the group? Since the VGDM may be inaccurate in detecting virtual groups, individual end points may be missing from the virtual group but yet be found later on. Thus, one must address how to distinguish inaccuracy versus the client actually no longer receiving data from the server application.

Due to the fact that there is no explicit multicast signaling outside of the domain, the makeup of the group itself must be derived from the monitoring of virtual group behavior over time. To that end, we propose to use a windowed history for each end client from the source application. The history window contains a sequence of N binary values whereby a 1 dictates that the client was included in the virtual group and a 0 means the client was absent. Since the virtual group is tracked on the basis of clients (destination IPs) internally rather than egress points, the history is kept on a source-wise basis. A state machine for the egress point is presented in [8].

---

[3] The notion of a discovery protocol is beyond the scope of the paper.

Virtual Groups

Time Left   *Clients*

Time Left		Clients
0.51 ms	Data	1,2,3,10
0.65 ms	Data	1,2,4,5,6,7
0.75 ms	Data	4,5,6,8,9

Share same source application

Src: *129.74.5.4*
Port: *665 UDP*

Source State

Filter: Allow
Egress
Thresh: 30%

Cost: 6
Lock: No

Current Tree   *PIM-SSM*
224.25.5.24

A 1,5   B 2,3,4,6   C 7,8   E 10   D 9

Egress List

A	146.5.23.5	InTree
B	146.5.23.150	InTree
...		
E	146.5.23.20	Eval

Client List

1	125.43.5.6	1087	8/10	A
2	43.10.5.6	2024	7/10	B
...				
10	64.5.25.8	9873	1/4	E

Fig. 3. Virtual groups versus application state

In the event that the multicast tree can be updated instantly, such as with stateless multicast approaches [9], there is no risk for coherency issues. However, given that the multicast control messages need to flow across the domain and follow through with the join or leave procedure, special care must be taken to ensure that changes to the group are restricted. Thus, we employ the concept of a time-based lock whereby once a change is initiated, further changes to the application state are restricted until a sufficient amount of time has passed. In the event an acknowledgment of changes is required, a message from the egress point involved in the change may be solicited by the VGDM.

3.3 State Management

A related component to the transport issue is how to manage the unique portions of state associated with domain. While the transmission of a packet to the appropriate egress points handles the issue of intra-domain distribution, the issues of how to convert the packet back to unicast and where the packet should be converted need to be addressed. For UDP-based applications (the primary beneficiary of stealth multicast), only the destination port and destination IP need to be saved from the packet. For all other fields in the packet, the fields should be the same.

To solve this issue, two approaches may be employed. In the first and simplest approach, the unique pieces of state can be directly included in the packet after the layer 4 (UDP) header. To the core routers in the domain, the additional state information appears as application-level data. When the packet reaches an egress node, the egress node will examine the packet to determine the end destination IDs that it is responsible for and appropriately create the new unicast packets for transmission onwards. Note that it is critical to specify the correct egress node to ensure that multiple egress points do not convert the same end destination IDs.

In the other approach, state information may be distributed amongst the egress points. Using a reliable transport, the VGDM can place the state infor-

mation at the egress nodes and use only a token to reference the actual state information rather than all of the unique portions of the data. A token must be included to correctly identify end clients as the actual recipients of the information may differ from the stored information. For applications with a semi-static list of clients, the distributed state approach presents an optimal solution as it minimizes the overhead for the transmission of state information. However, the approach does add additional complexity and state storage requirements at the edge router that may not be ideal for all circumstances.

3.4 Other Issues

In addition to the earlier design issues, we address several other issues associated with stealth multicast and its relevance that include the following:

– *Scalability:* Although there are potential concerns in stealth multicast with regards to scalability, the scalability concerns can be mitigated in several respects. First, the ability to correctly detect identical packet payloads at significant line speeds has already been documented via COTS hardware in [2]. Second the amount of storage required at the VGDM is limited due to the fact that only traffic that is likely to be multicast will be queued and the fact that only the first packet in a group needs to be completely stored. In [8] we demonstrate the low requirements for stealth multicast.
– *Practical benefit:* While stealth multicast is well suited for areas with a reasonable amount of redundant traffic, it is not envisioned that VGDMs become ubiquitous at all edge routers. Rather, it is envisioned that VGDMs will be placed at strategic locations in the network rather than via extensive distribution.
– *TCP:* While TCP traffic can work with stealth multicast it is unlike UDP, which is a connectionless send and forget mechanism. TCP involves the possible retransmission of dropped packets and additional state overhead. These features cause TCP traffic, in general, not to be amenable to stealth mulitcasting.

4 Simulation Studies

In this section, we present simulation studies of the stealth multicast architecture. The simulations were conducted using the ns-2 simulator. The purpose of our studies was to examine the performance implications of stealth multicast with regards to other approaches (ALM, unicast, IP multicast) as well as the impact on end user QoS.

The rationale behind our simulations was the following. In the network, Company X hosts an on-line gaming service with applications serving up to 64 clients. The applications are hosted on a set of servers with the clients existing throughout the global Internet. The parameters of the simulation were as follows:

– Random ISP domain (32 core nodes, 16 edge nodes).

- Server Farm (40 source Applications).
- The average number of clients listening to a server application was 32 clients randomly distributed amongst the edge nodes.
- Each join or leave (client joining or leaving) was exponentially distributed with an average inter-arrival event time of 500 ms for all clients in the simulation. The probability of the event being a join or leave was 0.5. The join/leave of the client was not conveyed to VGDM and observations on joins and leaves of the clients were derived using traffic patterns.
- The server applications sent data using UDP packets with an exponentially distributed packet rate of 50 ms and a packet size exponentially distributed with a mean of 500 bytes. For simplicity, the packets were streamed using a constant size unique to each application.
- The settings for the VGDM are listed in [8]

The primary purpose of the simulations was to evaluate the basic principles of the stealth multicast model (impact of queuing, predictability of control parameters, etc.). For the simulations, the bandwidth utilization and end-user QoS were used as the performance metrics. In the simulations, we compared the performance of four distinct models under varying configurations that included:

- *Traditional Unicast:* In this model, no stealth multicast is employed.
- *Full Stealth:* In this model, the VGDM is placed at the edge router of the ISP. Traffic must first pass through the customer's uplink before being considered as a candidate for stealth multicast.
- *Local Stealth:* In this model, the VGDM is placed at the edge router of the customer. The traffic can be considered for stealth multicast before being transmitted on the customer uplink.
- *ALM:* A generic version of ALM was used that is loosely based on End System Multicast [10]. Clients for ALM have an asymmetric bandwidth restriction with the ability to support 5 successive downstream connections.
- *IP multicast:* Everything is multicast that can be and no extra overhead B/W or latency is incurred.

All simulation results were normalized to IP multicast in order to compare the relative performance of the competing multicast technologies. In Fig. 4 unicast B/W consumption reaches its peak at 56 clients, this is due to packet loss starting to occur due to the link becoming saturated. Additional simulations can be found in [8].

4.1 Effect of Client Subscriptions, with Aggregation

In this simulation, background traffic (traffic not amenable to stealth multicast) is included. If the performance of stealth multicast is severely affected by background traffic than stealth multicast would be of little practical benefit. The background traffic was generated with simulated UDP and TCP sources. The sources were placed in such a way that all of their packets were detected by the
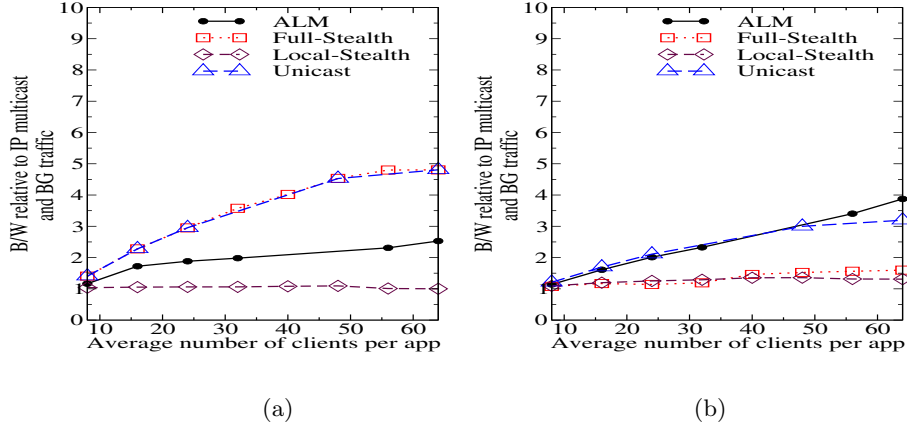
Fig. 4. Effect of average number of clients with Aggregation on (a) bandwidth - uplink (b) bandwidth - domain

VGDM. Figures 4 and 5(a) shows the performance of the VGDM under these conditions. As can be seen, stealth multicast offers a 3x to 4x improvement over the unicast case. The reason the difference between unicast and stealth multicast is lower is due to the inclusion of the non-multicast traffic. As this traffic is increased the performance ratio will necessarily go down.

Since the VGDM does only queues packets that are amenable to stealth multicast, these results are expected. However, it is possible that background traffic will spread out the arrival of redundant packets sufficiently enough in order to avoid detection by the VGDM. In this case the performance of the VGDM would degenerate into the unicast case, as no packets would be queued and all would be forwarded as standard unicast transmissions. The solution, as noted earlier, is to place the VGDM closer to the sources that are highly amenable to multicast.

4.2 Effect of Detection Parameters

Figure 5(b) shows the effect on queuing delay for the MHT setting. In this simulation PSW and TSW were disabled, allowing for the effect of MHT on the system to be measured. As expected the graph shows a slight linear increase in QoS Delay (it is only a small increase due to unavoidable delay being 33ms and stealth multicast only adds about 1 - 2 ms of delay on average).

However, normally MHT, PSW, and TSW work together in order to minimize the impact on QoS stealth multicast has. This introduces an important point for stealth multicast. In order for the aggregation to have a noticeable impact on queuing delay, the redundant packets need to be sufficiently dispersed so as to continually re-trigger the close proximity rewards of PSW and TSW.
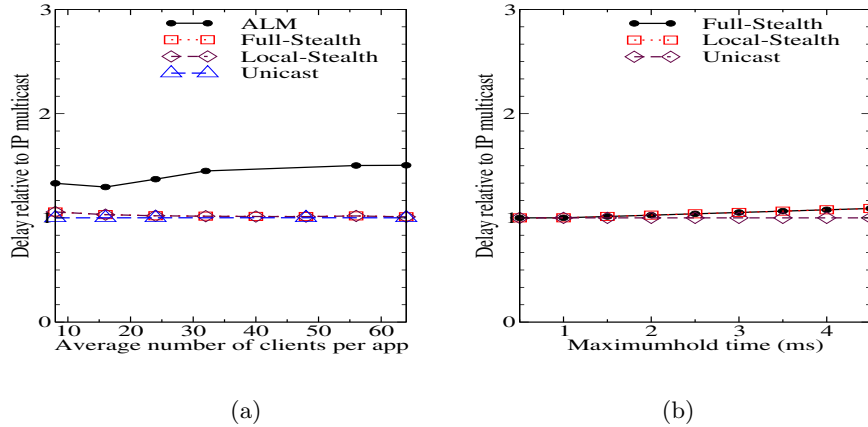
Fig. 5. (a) Effect average number of clients on QoS delay (b) Effect of MHT on QoS delay

5 Related Work

The most closely related work to stealth multicast lies in [2], whereby caching is applied at the packet level rather than the object/application level. However, unlike stealth multicast, packet caching fares poorly when the redundant traffic exhibits close temporal proximity (such as with streaming media). Furthermore, the work relies on unicast for transport whereas stealth multicast employs detection of virtual groups for multicast transport. Gathercast [11] is another technology related to dynamically increasing the efficiency of the network. However, it deals with a many-to-one (a reverse multicast) relationship where they combine many small packets (such as ACKs) that are for the same destination into one large packet, thus eliminating some overhead.

A short paper presenting the theoretical ideas for stealth multicast was presented in [12]. However this paper moves beyond our initial concept of stealth multicast in several key ways. First, the paper in [12] used DSMCast with tree encapsulation as the method for multicast transport. This caused significant overhead to the data packet to be introduced, thus limiting the benefit of stealth multicast. By using dynamic multicast trees at the VGDM and PIM-SSM signalling as the method for multicast transport, the need for encapsulated state information is removed. This gives a 3x savings in bandwidth which will only increase as the group size increases. Additionally, by using the background analysis engine in order to determine if a packet is likely to be amenable to stealth multicast, the need to queue all packets is removed. This eliminates the penalty normal unicast transmissions would incur in terms of QoS delay while also reducing the memory requirements for the VGDM. While the work in [13] is complementary in that the techniques can be applied to reduce the number of dynamic groups utilized, the work is significantly different in that it assumes an overall IP multicast framework rather than the dynamic conversion of stealth multicast.

6   Conclusions

Stealth multicast offers a novel approach for delivering the beneficial aspects of multicast with the deployment ease associated with cache-oriented approaches. By keeping the presence of multicast hidden from the external Internet, the key problems that have plagued network-level IP multicast are avoided. We believe stealth multicast offers a critical catalyst for spurring the development of large scale group-oriented applications that cannot occur in the current network environment. Stealth multicast offers a controllable and measurable economic benefit for ISPs to incorporate multicast-like efficiency without the complexities traditionally associated with multicast.

With regards to future work, we are developing an open-source prototype of the stealth multicast framework to run experimental studies on live traffic.

References

1. Danzig, P., et al.: A case for caching file objects inside internetworks. In: Proc. of ACM SIGCOMM'93. (1993)
2. Santos, J., Wetherall, D.: Increasing effective link bandwidth by suppressing replicated data. In: Proc. of USENIX. (1998) 213–224
3. Almeroth, K.: The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment. IEEE Network **14** (2000) 10–20
4. El-Sayed, A., Roca, V., Mathy, L.: A survey of alternative group communication services. IEEE Network (2003)
5. Mogul, J., et al.: Potential benefits of delta-encoding and data compression for http. In: Proc. of ACM SIGCOMM'97. (1997)
6. Wills, C.E., Mikhailov, M.: Towards a better understanding of Web resources and server responses for improved caching. Computer Networks (Amsterdam, Netherlands: 1999) **31** (1999) 1231–1243
7. Bhattacharyya, S.: An overview of source-specific multicast (ssm). RFC 3569 (2003)
8. Salyers, D., Striegel, A.: A novel approach for transparent bandwidth conservation. Technical Report TR-04-28, Univ. of Notre Dame Comp. Sci. and Engr. (2004)
9. Striegel, A., Manimaran, G.: DSMCast: A scalable approach for diffserv multicast. Computer Networks **44** (2004) 713–735
10. Chu, Y., Rao, S.G., Seshan, S., Zhang, H.: A case for end system multicast. IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast (2002)
11. Badrinath, B., Sudame, P.: Gathercast: The design and implementation of a programmable aggregation mechanism for the internet. In: Proc. of IEEE Int'l Conf. on Computer Communications and Networks (ICCCN). (2000)
12. Striegel, A.: Stealth multicast: A catalyst for multicast deployment. In: Proc. of IFIP Networking, Athens, Greece (2004)
13. Cui, J.H., Maggiorini, D., Kim, J., Boussetta, K., Gerla, M.: A protocol to improve the state scalability of source specific multicast. In: Proc. of IEEE GLOBECOM, Taiwan (2002)