

# A Novel Approach for Transparent Bandwidth Conservation

David Salyers, Aaron Striegel  
 Department of Computer Science and Engineering  
 University of Notre Dame  
 Notre Dame, IN. 46530 USA  
 Email: *dsalyers@nd.edu*, *striegel@cse.nd.edu*

**Abstract**—As broadband Internet access becomes more widespread, demand for rich multimedia content will increase significantly. When this trend is coupled with the point-to-point nature of the Internet, it is only natural that there will be an increase in redundant network traffic. Hence, a wide variety of techniques have been developed to eliminate redundant traffic in the network ranging from multicast to caching. Multicast techniques suffer from deployment issues, while caching techniques offer limited benefit for data with close temporal proximity (i.e. streaming).

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.

The features of **stealth multicast** allow it to be deployed without requiring changes to the server, client, or external Internet. Furthermore, QoS impact is strictly limited due to controls and the fact that only packets that are known to be likely amenable to multicasting are queued. Finally, **stealth multicast** allows the economic benefit of multicast to be directable, giving ISPs an incentive to deploy multicast.

**Index Terms**—Multicast, Stealth Multicast, Multicast Deployment, Network Efficiency, Bandwidth Conservation

## I. INTRODUCTION

As the Internet has grown and evolved, the demands on the network have shifted from simple connectivity to more sophisticated demands such as requirements for quality of service (QoS). The need for QoS has arisen

from the fact that the bandwidth of the network is not unlimited and hence must be managed appropriately between the competing users of the network. While various architectures such as Differentiated Services (DiffServ) [1] and others [2] offer frameworks for how to provide QoS, an alternative but complementary approach for QoS is to increase the efficiency of the underlying network traffic (bandwidth conservation). 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 [3], [4].

Hence, a wide variety of techniques have emerged to increase the efficiency of the network and as a by-product, the QoS as well. The spectrum of approaches range from an active coupling of the network and application (IP multicast [5]–[7] and application-level multicast [8]) to completely transparent approaches (packet caching [4], [9] and media caching [10], [11]). In one extreme, multicast-based approaches rely on the application and network working together to maximize efficiency. Multicast achieves its efficiency by providing an alternative transport mechanism (the multicast tree) for the dissemination of data. By presenting an alternative transport to the application, multicast is able to exploit the close temporal proximity of redundant data (i.e. bursts of the same data to multiple users).

In contrast to multicast, cache-based approaches do not require the support of the applications or the network, but typically operate transparently to the applications themselves. Even in the case of application-specific caching, various subtleties of the application protocol are exploited (DNS re-direction, etc.) rather than changing the core application itself. Furthermore, cache-based approaches differ significantly in that they exploit long term redundancy of data (i.e. the same data is accessed multiple times) rather than short term redundancy (as in multicast).

While both approaches have their respective benefits, both approaches have weaknesses that limit their ef-

This research was supported in part by the National Science Foundation through the grant CNS03-47392.

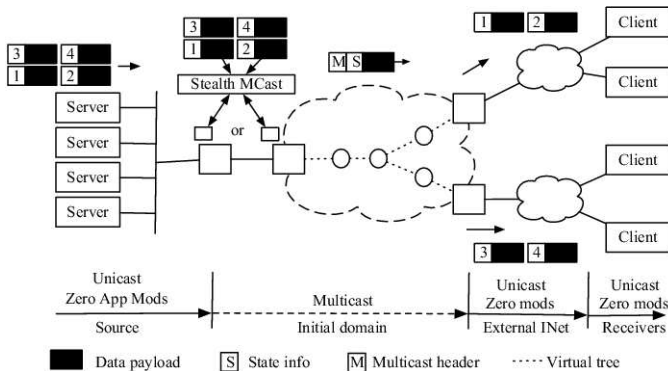


Fig. 1. Stealth Multicast - Overview

fectiveness. Most notably, multicast suffers tremendous deployment issues due to its global scale. Although techniques such as application-level multicast (ALM) offset this issue [12], this is traded for additional delay and a reliance on rich downstream clients. For cache-based approaches, the transparency helps with deployment but offers little or no benefit for redundancy with extremely close temporal proximity but no long-term redundancy.

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 close temporal proximity 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.

The remainder of our paper is organized as follows. Section II describes the concept of *stealth multicast*. Next, Section III describes the conversion process to multicast, the control of multicast tree construction, and other issues associated with *stealth multicast*. Section IV discusses the simulations performed and their results. Finally, Section V discusses related work and Section VI contains our concluding remarks.

## II. STEALTH MULTICAST OVERVIEW

Fig. 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<sup>1</sup>, 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 application transmits packets with the same data payload to different clients.
- 2) A digital signature is created for the data payload that uniquely identifies the data payload [9].
- 3) 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.
- 4) If the packet is to be queued, it is queued into a virtual group that shares the same attributes (packet size, signature, source IP, source port) as itself or a new virtual group is created.
- 5) Based on a predefined condition being met (group size, timer, etc.), the content of the virtual group is released to the transport mechanism.
- 6) If the group size is sufficient, the virtual group is transmitted via multicast after selecting the appropriate multicast tree. If not, the packets are forwarded using standard unicast transmission.
- 7) The unique portions of each packet are preserved as state information. The unique portions include the destination IP and destination port.
- 8) The packet is transported across the domain using the underlying transport mechanism (PIM-SSM [13] or unicast).
- 9) 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.

*Stealth multicast* can be used for a wide range of applications including applications that cannot take advantage of other current multicast approaches. One such appli-

<sup>1</sup>The virtual group itself may have little or no relation to the physical group (actual multicast tree).

cation is on-line games where end-to-end network-level multicast support is not available and/or the strict delay requirements do not permit application-level multicast. Other examples might include legacy applications whose server or client code cannot be modified. Additionally, the scale of these applications is only limited by the amount of memory required to queue the packets at the VGDM, which will be shown to be quite minimal later in the paper. Further scaling constraints such as the MTU (due to stealth multicast signalling overhead) may be overcome via partitioning virtual groups into multiple stealth multicast packets.

Critical to the ISP, stealth multicast is a sender-driven approach to multicast rather than the receiver-driven approach of network-level IP multicast. Rather than changes to the multicast group being controlled by the client or end system, 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; a value not easily divined with existing multicast approaches. Finally, 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 end-to-end transport, which improves deployability.

**Key Principles:** Throughout its operation, stealth multicast adheres to two key principles, namely *external transparency* and *negligible QoS impact*. The first principle, external transparency, is critical for several reasons, most notably for the issue of deployment. Stealth multicast changes the problem of multicast deployment to one of a domain-wise problem rather than a global problem by virtue of keeping the presence of multicast hidden from the global Internet. Since each domain can operate independently (conversions occur at the edge of domain), there are no issues with global interoperability or the majority of complexities associated with correct network-level multicast operation [5], [6], [14]. In fact, stealth multicast relies on readily available intra-domain multicast routing protocols and requires the addition of only a few control messages and processing at the edge routers for operation. The majority of the processing of stealth multicast can be accomplished via COTS hardware co-located with the ingress router for the domain [9].

The second principle ties into the first principle and into the stealth of the model itself. If the QoS of the user is significantly impacted in either the positive or the negative direction, the fact that stealth multicast is being employed may be discernible. A significant QoS change may impact the functionality of the applications utilizing

the network as well. Although a positive QoS impact may not necessarily generate criticism, a negative impact on QoS will certainly cause issues with application functionality. However, this principle has an inherent amount of flexibility due to the fact that only a ‘noticeable’ QoS impact causes any issues. Due to the fact that QoS is subject to both the perception of the end user and the requirements of the application, it is the prerogative of the network administrator to determine what constitutes a noticeable QoS impact. For our paper, we define the term *noticeable QoS impact* to refer to the end user attributing the poor network performance to something other than the typical variations in Internet traffic behavior.

**Additional Benefits:** First, 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 accuracy, 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, since all tree operations are driven by the VGDM and occur only on a domain-wise basis, the issues associated with resource management are vastly simplified. 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.

### III. 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. Fig. 2 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 [4], [9], which will give a checksum that is sufficient in

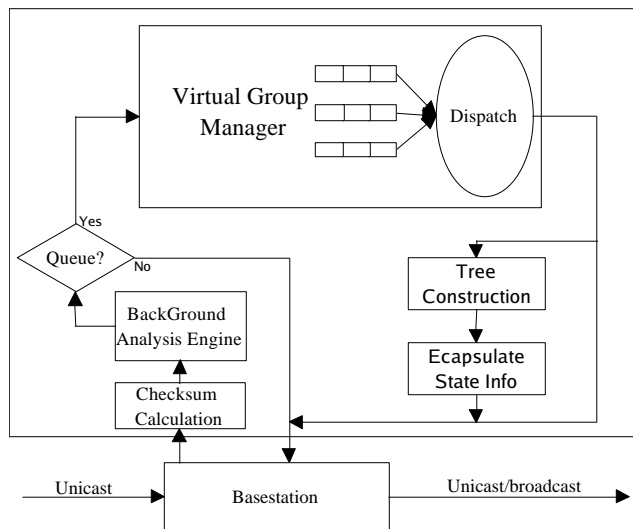


Fig. 2. Stealth multicast module - basic components

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. This engine is used to track if a specific source IP and port is likely to produce packets that will benefit from stealth multicast. This is determined by keeping track of if the data from a specific source IP and port would have been multicast 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.

The VGDM has a limited number of slots that can be used to store the information required for the analysis engine. Due to this limitation, a replacement policy is needed for the entries in the analysis engine. When a new entry needs to be created and there are no vacant slots, the entries are scanned for the first entry for a source that is known not to amenable to stealth multicast. The first

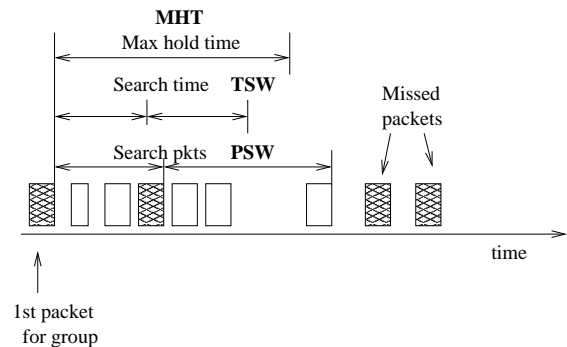


Fig. 3. Virtual group search settings

entry that is found is then replaced with the new entry. If all entries are listed as being amenable to multicast then the timestamp of last access is checked until it finds the first entry with that has been unused for a specified amount of time. This entry is then replaced with the new entry. If no suitable location for the new entry can be found then no entry is created and the packet will not be transmitted by multicast.

#### A. 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 (queues). The data payloads are uniquely identified by the signature, the packet size, and application source (source IP, source port). In addition, other potential information for each packet such as the DS field (for DiffServ networks) may also be preserved on a packet-wise basis. The packets themselves are queued in the virtual group until an appropriate trigger causes the virtual group to be dispatched.

The triggers for dispatch define the performance (additional multicast efficiency) as well as the penalty (additional queuing delay) introduced by stealth multicast. The goal is to balance the impact of queuing delay versus the potential benefit derived from the detection of additional redundancy for the virtual group. For instance, if packets are held too long, the end-to-end delay increase may be noticeable by the end user. Again, since packets from sources that are not known to benefit from multicast are not queued, the impact stealth multicast has on standard unicast transmissions is negligible. The following parameters govern the triggers for release of virtual groups (see Fig. 3):

- **PSW - Packet Scan Width:** The maximum number of packets to scan before the virtual group is released. In the event of a new addition to the virtual group, this counter is reset.

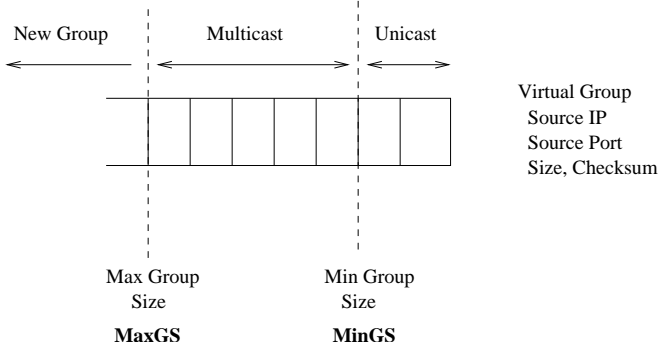


Fig. 4. Virtual group threshold settings

- ***TSW*** - *Time Scan Width*: The amount of time to scan before the virtual group is released. Similar to the *PSW*, this timer is reset upon the addition of a new packet to the virtual group.
- ***MHT*** - *Maximum Hold Time*: The maximum time that a virtual group can exist and hence the maximum time that a packet can sit inside a queue. This timer is set when the virtual group is started (first packet) and places a bound on the queuing time, regardless of additions to the virtual group.

Once a packet is triggered for release, it is given to the transport mechanism for dispatch via multicast or unicast (insufficient virtual group membership). Fig. 4 shows the effects of the group threshold settings that are discussed in more detail below:

- ***MaxGS*** - *Maximum Group Size*: This parameter forces a group to be dispatched when it passes a specific size. The intuition is that the group has passed a necessary efficiency and should be dispatched as soon as possible.
- ***MinGS*** - *Minimum Group Size*: In order to be considered as a candidate for multicast, a group must meet a certain minimum membership level. For extremely small virtual groups, it may be simpler to employ separate unicasts.

In short, *PSW* setting governs the search width from the perspective of flow aggregation / virtual group detection (i.e. how mixed is the packet in the incoming aggregate flow) whereas the *TSW* setting reduces queuing time in the event of idle inputs to the VGDM. Intuitively, the closer that the VGDM is to the source, the tighter the *PSW/TSW* values that can be employed. If the VGDM is located extremely close to the traffic source, there is a much better chance that additional redundant packets will be located close together. As the VGDM is placed farther away from the source, there is a decreasing chance for group detection due to the fact that the group traffic may be interspersed with traffic

from other sources or applications. The search width may also be affected by the OS sending the packets as well as the scheduling mechanisms (application, shaping, etc.) leading up to the VGDM.

The *MHT* setting allows the network administrator to bound the delay impact of virtual group detection. The *MHT* places a worst case delay on the virtual group detection while the actual delay experienced will depend upon the underlying traffic patterns and the *TSW/PSW* settings. While one could operate using only *MHT*, the *PSW* and *TSW* allow the virtual group manager to quickly empty out non-growing (and hence non-contributing) virtual groups.

**Overflow:** In addition to considering the user QoS impact of the additional queuing introduced by stealth multicast, the system is also subject to the constraint of the memory of the VGDM (i.e. the number packets that can be successfully queued). The worst-case constraint for the system can be described as follows:

$$\frac{LR \times MHT}{MinGS} + \left( \frac{LR \times MHT}{MinGS \times MinPktSize} \right) OH$$

where *LR* is the line rate entering the VGDM, *MHT* is the maximum hold time, *MinPktSize* is the minimum packet size for the link, and *OH* is the storage overhead related to each virtual group. We assume that  $TSW = MHT$  and  $PSW = \frac{LR \times MHT}{MinGS \times MinPktSize}$  such that *TSW* and *PSW* are not a factor and that no packets are discarded due to filtering. In the worst case, there will only be *MinGS* clients for each virtual group and packets are held for the *MHT*. Since *MHT* is inherently limited by the end impact on QoS, the actual storage requirement is also limited. For instance, a 1 Gb/s link with a 5 ms *MHT*, a *MinGS* of 2, and an overhead of 1000 bytes per group would require roughly 6 MB of storage space.

However, the storage required by the VGDM will be significantly less due to two reasons. The first is, a source has to be known to benefit from multicasting before data coming from the source will be queued, otherwise it is immediately forwarded and not stored in the VGDM. The second reason the storage requirements are kept low is that each virtual group is only held for *MHT*, starting from the arrival time of the first packet in the group. This means that most packets will not be held for the whole *MHT* time. Generally, as *LR* increases *MHT* should decrease in order to prevent an overflow of the VGDM and to lessen the impact on QoS.

Finally, while overflow may occur in the system, an overflow of the VGDM will not result in a critical failure or direct loss of data. In the case that insufficient space

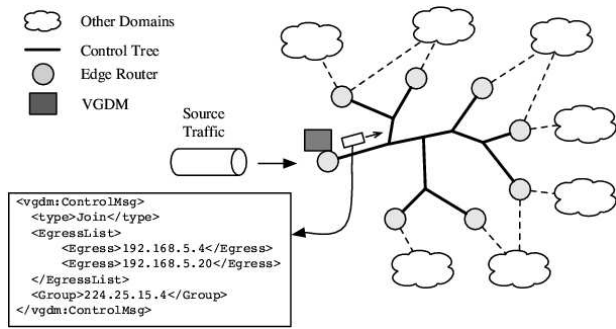


Fig. 5. VGDM Control Messages

is available at the VGDM, an overflow will simply result in the packet sent onwards as a unicast packet. Hence, the efficiency gains of the VGDM will not be available until the overflow condition is removed.

### B. 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). In addition, the egress point may contain the replication for multiple downstream clients, thus further reducing scalability issues.

Although a variety of approaches can be employed to manage multicast transport across the domain, the selected approach must balance the simplicity of the transport mechanism versus the efficiency of the solution. On one extreme, the approach of broadcasting packets and releasing the packets later (broadcast/hold) offers a simple solution at the cost of inefficiency with smaller or non-uniformly distributed egress distributions. To improve efficiency while avoiding group dynamics, the next step is to offer a grouping of fixed trees to capture multiple combinations of egress points. While this problem is relatively simple during operation, the construction of an optimal tree that can be traversed quickly is a difficult problem that is an open topic for future research.

Thus, 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 Fig. 6). Unlike the broadcast approach whereby all egress points are on the tree, 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<sup>2</sup>), 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 Fig. 5). 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

<sup>2</sup>The notion of a discovery protocol is beyond the scope of the paper.

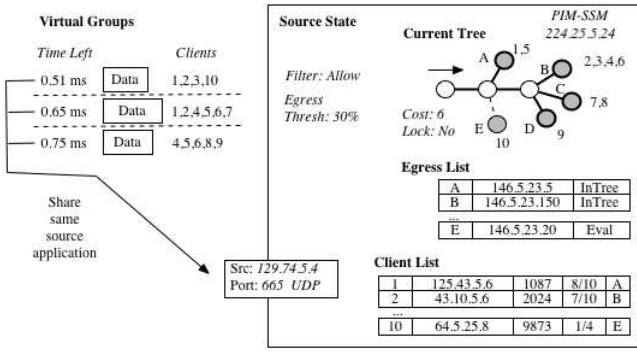


Fig. 6. Virtual groups versus application state

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.

For each of the egress points that cover the clients of the source application, a state value is maintained (see Fig. 7) to denote whether or not the egress point is part of the actual multicast tree<sup>3</sup>. When the client is first noted and its egress point is new, the egress point state is set to the *Eval* state. Once the client has demonstrated an appropriate level of consistency, denoted by  $X$  out of  $N$  successful detections, the egress state transitions to the *InTree - Join* state. A *Recalc* flag is set for the application state to denote that the multicast group should be updated.

To keep the multicast group covering only valid egress points, the list of egress points is checked in a periodic manner for egress points to purge from the list. During evaluation, all egress points in the *InTree* state are set to *InTree - Verify* and transitioned back to *InTree* if client histories exist with a sufficient threshold to justify continued inclusion in the multicast group. Note that only one client is required to justify an egress point as multiple clients may share the same egress point. If the egress point stays in the *InTree - Verify* (i.e. no clients need the egress point), the egress point is transitioned to the *Purge* state and the *Recalc* flag is set to trigger an update of the tree. The purging of *Eval* nodes does not require a change to the tree as the egress point has not yet been officially added.

In the event that the multicast tree can be updated instantly, such as with stateless multicast approaches

<sup>3</sup>It is assumed the VGDM can receive and interpret BGP messages in order to derive domain egress points for the network.

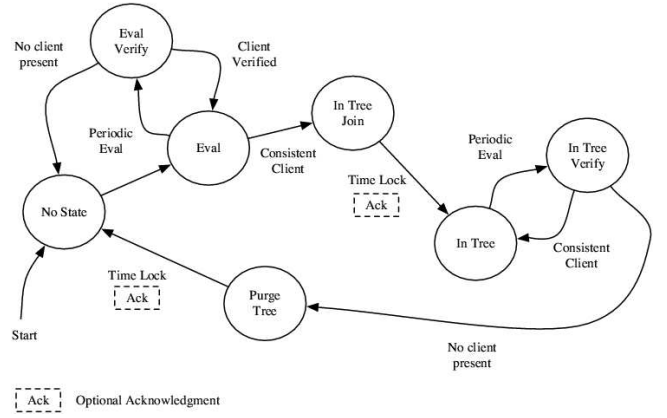


Fig. 7. State machine for an egress point

[15], [16], 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.

### C. 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 information 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.

#### D. 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 [9]. 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.
- *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. While the predicted level of multicast traffic can vary significantly across the networking community (minimal to significant), our preliminary investigations of the local Internet link where group applications are run, indicate that a 5% to 15% gain can be realized immediately with only a basic VGDM. With support for TCP-based applications and local users recognizing that group-oriented applications no longer endure a linear bandwidth bottleneck at the local Internet uplink, we believe that a gain of 20% to 40% and even beyond can be realized.
- *IPv6:* Beyond the longer address space associated with IPv6 and its implications for state management, IPv6 will not have a large impact on stealth multicast. The most important effect on stealth

multicast will be the forced adoption of distributed state at the egress routers to adjust to the extended IP address length.

- *ALM:* Although ALM may appear to be contradictory to stealth multicast, ALM-based applications can still benefit from stealth multicast. Provided that the actual data payload is still identical, stealth multicast will make no distinction between ALM and non-ALM traffic. Given the typical environment of asymmetric broadband access to most users (high download, small upload), stealth multicast can help alleviate the lack of rich users who can significantly contribute to other users by offsetting costs at the source. In fact, one could develop an ALM-aware VGDM to further introduce additional gains for the two technologies.
- *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 multicasting.

## IV. SIMULATION STUDIES

In this section, we present simulation studies of the stealth multicast architecture using the scenario depicted in Fig. 8. The simulations were conducted using the ns-2 simulator [17] and the GenMCast extension module [18]. 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 analysis domain for stealth multicast is the domain immediately following the hosted servers with the intermediate domains between the initial ISP and the end clients ignored. The parameters of the simulation were as follows:

- The simulations were conducted on a random ISP domain with a topology consisting of 32 core nodes and 16 edge nodes.
- The server farm consisted of 10 servers with each server hosting 4 separate applications for a total of 40 separate server sources.
- 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-

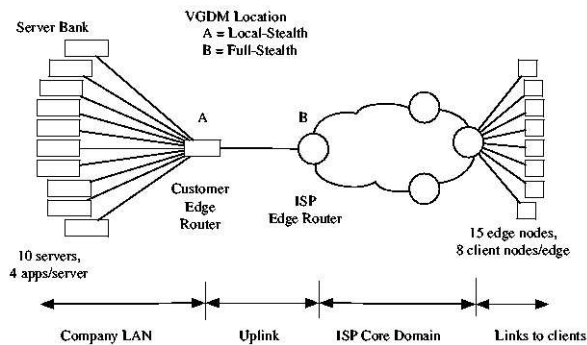


Fig. 8. Network Simulation Layout

TABLE I  
VGDM SETTINGS

Parameter	Setting
Maximum Groups	50
Maximum Hold Time (MHT)	5 ms
Time Search Width (TSW)	2 ms
Packet Search Width (PSW)	100
Min Group Size (MinGS)	2
Max Group Size (MaxGS)	200
State Management	Distributed
Egress Threshold	30%
Time Lock	100 ms

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.
- A PIM-SSM capable network was assumed for the ISP network.
- The settings for the VGDM are listed in Table I.

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 following metrics were used to evaluate the performance of stealth multicast:

- *Bandwidth utilization:* The bandwidth consumption of the server traffic on the uplink from Company X and the bandwidth cost on the initial ISP domain were examined.
- *End-user QoS:* The effects on end-user QoS were examined to determine the impact that the additional queuing delay of stealth multicast. The end-

to-end queuing delay for individual clients was compared versus other approaches to assess the relative impact of the delay.

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. This model is used as a baseline for comparing the performance of the other models.
- *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 [12]. 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 both Fig. 9 and Fig. 10 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.

#### A. Effect of Client Subscriptions - No Aggregation

The fundamental motivation for the stealth multicast model is to offer a significant bandwidth improvement in the core of the domain. Fig. 9 plots the performance as the average number of clients per server application is varied from 8 to 64, representing a typical medium scale application. As would be expected, the unicast-only model offers the worst performance in all cases. However, the most notable aspect is the performance of ALM versus stealth multicast. Despite the fact that ALM shifts much of the replication burden downstream to the end clients, the asymmetry of the clients incurs a significant penalty to the source, thus requiring it to send multiple copies. The performance gap is notable in both Fig. 9 (a) and (b). This performance gap would be eliminated by the inclusion of stealth multicast and ALM together.

Most notably, the actual queuing delay of the VGDM is quite minimal as shown in Fig. 9(c). Unlike ALM

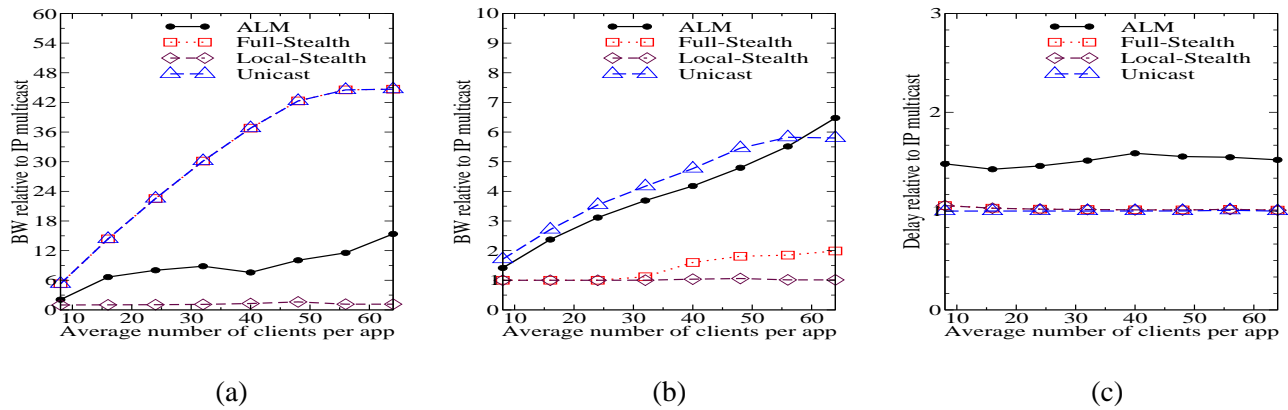


Fig. 9. Effect of average number of clients on (a) bandwidth - uplink (b) bandwidth - domain (c) end-to-end queuing delay

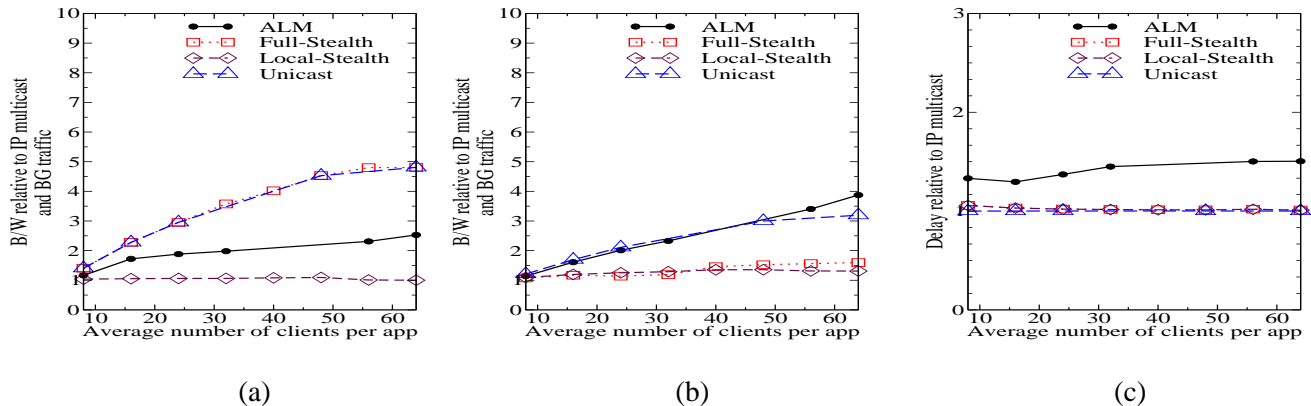


Fig. 10. Effect of average number of clients with Aggregation on (a) bandwidth - uplink (b) bandwidth - domain (c) end-to-end delay

which adds additional delay due to a longer distribution tree, stealth multicast adds a barely perceptible 1-2 milliseconds of delay to the end-to-end delay. In fact, with the distance between many client nodes in the simulation being relatively small (10 ms), one can reasonably infer that the end-to-end delay of ALM would decay even further with additional distance between the client nodes. Most important of all, stealth multicast can offer significant bandwidth improvements with zero modifications to the client or server applications ranging from 5x over unicast over the domain to over 45x for the uplink in the local VGDM case. The differences in performance gain between the uplink and domain can be attributed to disjoint links (the uplink is always shared while the links in the domain may need to be traversed regardless) and inaccuracies in the actual multicast group (clients have already joined or left the group).

### B. Effect of Client Subscriptions, with Aggregation

The previous section dealt with the affect the number of client subscriptions had on the stealth multicast. However, for those simulations there was no background traffic that was not amenable to multicast. Thus, stealth

multicast would be of little practical benefit if its performance was affected by increasing the background noise. 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 VGDM. Fig. 10 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.

These results were as expected due to the fact that the VGDM does not queue all packets, and in fact, will not queue packets until the source IP and port combination are determined to be a good candidate for stealth multicast by the background analysis engine. However, it is conceivable that significant background traffic could exist such that it would spread out the arrival of packets that could benefit from multicast enough so that the packets no longer arrive in the performance constraints of the VGDM. If this occurs the performance of the VGDM would degenerate into the unicast case, as

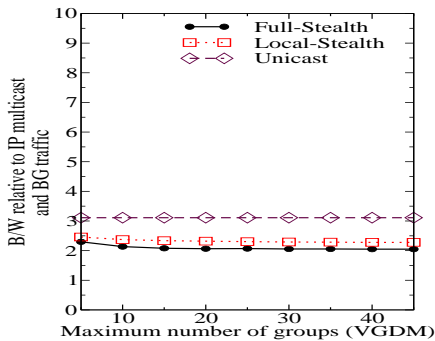


Fig. 11. Effect of maximum number of virtual groups (VGDM size) on the domain bandwidth

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.

### C. Effect of System Constraints - Maximum Groups

In order for the VGDM to function properly, there must be sufficient queuing space available at the VGDM to hold new virtual groups. However, the *MHT* and the fact that only traffic that is likely to benefit from stealth multicast is queued keeps the storage space requirement low. As can be seen in Fig. 11, once the number of virtual groups reaches the number of multicast servers there is little gain to be made from adding more virtual groups. This implies that to get the best performance the number of multicast groups that the VGDM can store should be slightly greater than the number of sources likely to serve traffic that would benefit from multicast. However this has a hard limit due to the line rate of data coming into the VGDM and the minimum group size.

If there are not enough virtual groups in the VGDM to hold all of the traffic that could be multicast the system does not fail. Traffic that can not be queued due to the virtual group limit being reached is immediately forwarded as a unicast packet.

### D. Effect of Detection Parameters

Figures 12 and 13 shows the effect on queuing delay for both the *TSW* and *MHT* settings. In the simulation showing the effect of *TSW*, *PSW* and *MHT* were disabled, and for the *MHT* simulation *TSW* and *PSW* were disabled. This allows the measurement of the individual effects on QoS each of these settings have. As expected both graphs show a slight linear increase in QoS (it is slight due to unavoidable delay being 33ms and stealth multicast only adds about 1 - 2 ms of delay on average).

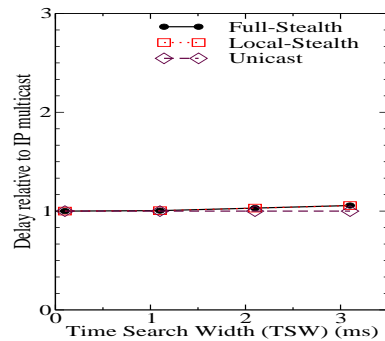


Fig. 12. Effect of the packet search width (TSW) on average end-to-end delay

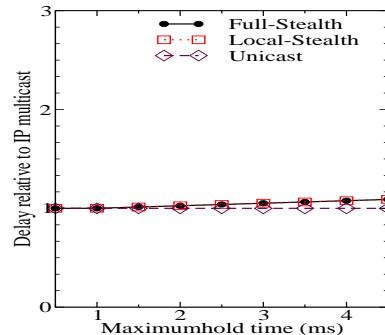


Fig. 13. Effect of the maximum holding time (MHT) on average end-to-end delay

However, these three settings normally 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*. For both the *Full-Stealth* and *Local-Stealth* cases, the VGDM is placed relatively close to the source, thus avoiding much of the intra-flow jitter that is introduced as the packets travel farther downstream. If the VGDM were placed on an inter-domain link to reduce traffic from other domains, the *PSW* and *TSW* would not be able to be set as tightly. With too tight of a bound, the VGDM would miss potential candidates due to more significant delays between redundant data packets.

## V. RELATED WORK

The most closely related work to stealth multicast lies in [4], [9], 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.

A short paper presenting the theoretical ideas for stealth multicast was presented in [19]. However this paper moves beyond our initial concept of stealth multicast in several key ways. First, the paper in [19] 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 [20] 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.

## VI. 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. Additionally, stealth multicast is complementary to ALM by helping to negate the impact of asymmetric clients, an important quality for improving the scalability of delay-sensitive ALM-based applications.

Since stealth multicast will only queue traffic from sources that are known to be amenable to stealth multicast no additional queuing delay is added for standard unicast traffic. Additionally stealth multicast offers absolute controls for limiting the impact on user QoS for those packets that do get queued. 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] K. Nichols, S. Blake, F. Baker, and D.L. Black, "Definition of the Differentiated Services field (DS Field) in the IPv4 and IPv6 headers," *IETF RFC 2474*, Dec. 1998.
- [2] R. Braden, D. Clark, and S. Shenkar, "Integrated Services in the Internet architecture: An overview," *IETF RFC 1633*, June 1994.
- [3] P. Danzig and et al., "A case for caching file objects inside internetworks," in *Proc. of ACM SIGCOMM'93*, 1993.
- [4] J. Santos and D. Wetherall, "Increasing effective link bandwidth by suppressing replicated data," in *Proc. of USENIX*, 1998, pp. 213–224.
- [5] K. Almeroth, "The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment," *IEEE Network*, vol. 14, pp. 10–20, Jan./Feb. 2000.
- [6] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for IP multicast service and architecture," *IEEE Network*, pp. 78–89, Jan./Feb. 2000.
- [7] M. Ramalho, "Intra- and Inter- domain multicast routing protocols: A survey and taxonomy," *IEEE Communications Surveys and Tutorials*, vol. 3, no. 1, pp. 2–25, Jan.-Mar. 2000.
- [8] A. El-Sayed, V. Roca, and L. Mathy, "A survey of alternative group communication services," *IEEE Network*, Jan./Feb. 2003.
- [9] N. T. Spring and D. Wetherall, "A protocol independent technique for eliminating redundant network traffic," in *Proc. of the 2000 ACM SIGCOMM Conference*, Stockholm, Sweden, Aug. 2000.
- [10] J. Mogul and et al., "Potential benefits of delta-encoding and data compression for http," in *Proc. of ACM SIGCOMM'97*, 1997.
- [11] Craig E. Wills and Mikhail Mikhailov, "Towards a better understanding of Web resources and server responses for improved caching," *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 31, no. 11–16, pp. 1231–1243, 1999.
- [12] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, Oct. 2002.
- [13] S. Bhattacharyya, "An overview of source-specific multicast (ssm)," *RFC 3569*, July 2003.
- [14] R. Beverly and K. Claffy, "Wide-Area IP Multicast Traffic Characterization," *IEEE Network*, Jan./Feb. 2003.
- [15] R. Boivie, "A New Multicast Scheme for Small Groups," *IBM Research Report RC21512*, June 1999.
- [16] A. Striegel and G. Manimaran, "DSMCast: A scalable approach for diffserv multicast," *Computer Networks*, vol. 44, no. 6, pp. 713–735, Apr. 2004.
- [17] "UCB/LBNL/VINT Network Simulator - ns (version 2)," Available at [www.mash.cs.berkeley.edu/ns/](http://www.mash.cs.berkeley.edu/ns/).
- [18] "GenMCast: A generic multicast extension for ns-2," Available at [www.cse.nd.edu/striegel/GenMCast](http://www.cse.nd.edu/striegel/GenMCast).
- [19] A. Striegel, "Stealth multicast: A catalyst for multicast deployment," in *Proc. of IFIP Networking*, Athens, Greece, May 2004.
- [20] J. H. Cui, D. Maggiorini, J. Kim, K. Boussetta, and M. Gerla, "A protocol to improve the state scalability of source specific multicast," in *Proc. of IEEE GLOBECOM*, Taiwan, Nov. 2002.