Stealth Multicast: A Novel Catalyst for Network-Level Multicast Deployment

A. Striegel

Systems & Software Laboratory
Dept. of Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46530 USA
E-mail: *striegel@cse.nd.edu*

Abstract. While network-level multicast has tremendous potential for increasing the efficiency of group-oriented applications, the adoption of network-level multicast has been tepid at best. In this paper, we propose a novel concept entitled stealth multicast that allows for practical adoption of network-level multicast on a domain-wise basis rather than global scale. In the stealth multicast model, similar unicast packets are dynamically assembled into virtual groups for multicast transmission across the domain. At the edge of the domain, the packets are converted back to unicast, thus hiding the existence of stealth multicast from the external Internet. True to its namesake, stealth multicast operates in complete stealth, providing seamless interoperability without requiring any modifications to end-user applications nor requiring any inter-domain support. In this paper, we introduce the basic concepts of stealth multicast and show that the stealth multicast model can offer significant benefits in terms of bandwidth savings with minimal impact to the end-user QoS.

Keywords: Multicast Deployment, Routing, QoS

1 Introduction

While fundamental multicast concepts have been successfully deployed in the Mbone and exist in many commercially available routers, recent studies show a relatively lackluster adoption over the last decade [1, 2]. Furthermore, despite the large body of research on network-level multicast [3], recent trends in multicast research have shifted to application-level multicast (ALM) [4]. Although techniques such as ALM can offer near network-level multicast bandwidth savings, ALM can suffer from additional delay due to longer distribution trees and a dependence upon a rich end-user capacity to provide adequate downstream branching. Despite its weaknesses, ALM offers a compelling solution for bandwidth management as it avoids one of the key problems associated with network-level multicast, namely global network deployment.

Whereas much can be written about the tepid adoption of network-level multicast, the root of the problem arises from the fact that much of the benefit of

network-level multicast only comes with complete global deployment. The challenge of global network-level multicast support appears especially daunting given that many other complex sub-issues also provide obstacles such as the support vs. development vs. demand dilemma, deployment complexity (billing, management), and ISP economic incentive [5]. Thus, our paper poses the following question, is it possible to offer a novel approach to multicast that allows for incremental deployment while avoiding the pitfalls that have plagued network-level multicast deployment (application adoption, ISP incentive, etc.)? This question provides the basis for the model proposed in our paper, *stealth multicasting*.

1.1 Stealth Multicast Overview

At its core, the stealth multicast model changes the context of the problem regarding multicast deployment. Rather than requiring participation on a global scale, stealth multicast abstracts multicast transport in an individual domain (Autonomous System) such that its entire presence operation is kept hidden from the outside world. At the edge of the domain, packets are dynamically converted to and from multicast, thus allowing for seamless interaction with existing unicast applications. The conversion for multicast is done only at inputs to the domain whereby maximum rewards can be gleaned through reduction to multicast transport. Hence, the target audience for stealth multicast is domains directly serving UDP applications that send out identical data streams to multiple users (i.e. streaming media, on-line games) that network characteristics that ALM cannot meet (minimal delay, reduced client capabilities, etc.).

The rest of our paper is organized as follows. In Section 2, we present the fundamentals of the stealth multicast model. Next, in Section 3, we present the MYDEKI model. Then, in Section 4 we conduct simulation studies regarding the performance of the MYDEKI model. Finally, in Section 5 we offer several concluding remarks and discuss our future work.

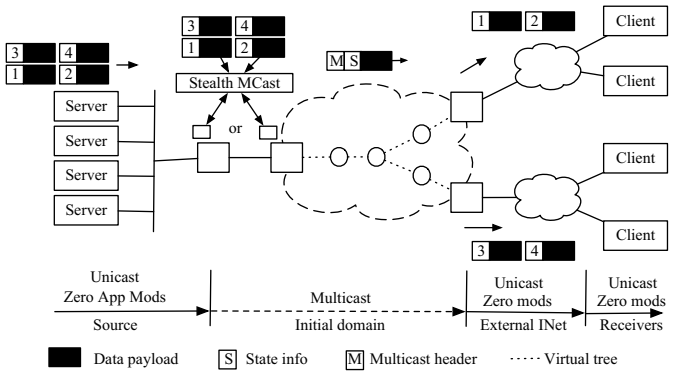


Fig. 1. The stealth multicast model

2 The Stealth Multicasting Model

In order to provide seamless interoperability with minimal end-system modification, the stealth multicast model relies on two governing principles that are as follows:

- *Externally transparent:* External unicast applications should not require any modifications in order to operate in the stealth multicast environment. The same applies for existing multicast applications (if present).
- *Negligible QoS impact:* The end-user should not experience a *noticeable* impact in QoS.

With the first principle of external transparency, the goal is to provide seamless functionality with existing networks. This principle is extremely critical due to the tremendous number of applications currently in use, many of which may be extremely difficult or even impossible to change. Note that this principle does not preclude future applications from taking advantage of the stealth multicast model, ALM, or or network-level multicast but rather ensures backwards compatibility with existing IP unicast applications.

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.

2.1 Stealth Multicast Operation

Figure 1 shows the overall concept of the stealth multicast model. The key component of the stealth multicast model is the Virtual Group Detection Module (VGDM) that is shown in Figure 2. The VGDM is placed at the edge of the domain and queues packets for assembly into *virtual groups* for multicast transport across the domain.

The stealth multicast process begins as a group-oriented application transmits packets via separate unicasts to multiple clients. The packets travel via the uplink to the domain and arrive at the edge router (see Figure 1). The packets are then transferred to the VGDM for virtual group consideration. A filter may be applied at the edge router to remove packets from consideration that should

not or would never become part of a virtual group. Examples of such packets would include existing multicast packets, ICMP, and RSVP.

Depending upon the behavior desired and involvement of the source application, the VGDM may be placed along the path at various points in the network. While placement at the edge of an ISP's domain such as in Figure 1 would allow for maximum stealth and benefit solely for the ISP, the VGDM could alternatively be placed at the edge of the customer LAN to decrease the load on the uplink to the domain. Furthermore, the VGDM could also be placed on an incoming link from another domain. However, as will be discussed later, the stealth and utility of such placement may be reduced depending upon the nature of the underlying traffic. 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 shows the steps involved once the packet arrives at the VGDM which are discussed below.

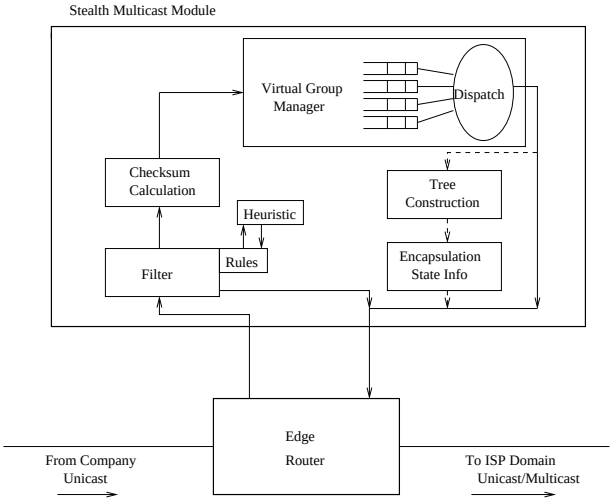


Fig. 2. Stealth multicast module - basic components

Initial Filter: Once a packet arrives at the VGDM, the first step is to apply a basic filter to the packet. The packet is filtered according to a set of rules as defined by the network administrator. The rules may also be generated by heuristics monitoring the incoming flows to optimize the candidates for consideration.

Checksum Calculation: If a packet passes the filter, it is uniquely categorized according to its data contents. The checksum calculation module creates a checksum that uniquely identifies the payload of the data packet. The checksum must be sufficient such that two unique payloads of the same size do not com-

pute to the same checksum. The checksum module is only interested in the data payload as the remaining header information (IP, UDP) is handled by the virtual group manager. The checksum (digital signature) is computed using COTS hardware [6].

Virtual Group Manager: Next, the packet is passed to the virtual group manager for placement into virtual groups (queues). The virtual groups are uniquely identified by the shared checksum, packet size, source IP, and source port. The packets themselves are queued in the virtual group until an appropriate trigger causes the virtual group to be dispatched. The triggers determine both the performance (additional multicast efficiency) as well as the impact on end-user QoS introduced by the VGDM.

Tree Construction: After a sufficient stimulus has occurred (time, size, etc.), the packets are given to the dispatch mechanism for transmission onwards. Providing that the virtual group has sufficient membership to justify the use of multicast transport, the packet is given to the tree construction module. If the size of the virtual group is not sufficient, the packets are simply released as standard unicast packets back to the edge router.

Encapsulation of State Information: One of the inherent problems in stealth multicast is that virtual groups are constructed dynamically, i.e. the makeup of the end clients is not know a priori. In addition, since it is assumed that multicast does not exist outside of the ISP domain, the packets must be converted back to unicast at the edge of the domain. However, using standard IP multicast which contains only the source IP/ destination group address, the edge routers would not know whom is responsible for sending the unicast packet nor the unique portions of the headers (DS field, destination IP, UDP destination port). Hence, additional state information must be included in the packet or kept at the egress routers to identify how the packet should be converted back to unicast.

Transmission & Exit: After the packet has been modified for multicast transport (tree construction) and the state information has been added (encapsulation), the packet is given back to the edge router. The edge router then forwards the packet across the domain using the underlying multicast transport mechanism (broadcast, route-pinning, ALM, etc.). Once the packet reaches the edge of the domain, the egress router is responsible for converting the packet back into a unicast packet. Using the virtual group state information, the egress router reconstructs the appropriate unicast packets and forwards the replicated unicast packets onwards. The packets are identical to the packets that were originally seen at the VGDM with appropriate modifications for TTL and any other necessary fields. Once the packet leaves the domain, it is a standard unicast packet, indistinguishable from any other unicast packets that did not undergo multicast consolidation/transport across the domain.

To both the application sending the packets and the client receiving the packets, there is no difference in the contents of the packets nor a noticeable change in the QoS of the packet. However, there is certainly a noticeable impact for the domain. In addition to avoiding upstream bottlenecks, the general bandwidth

requirements of the domain are alleviated due to the use of multicast. In fact, stealth multicast allows for a clear transition of deployment from the bare minimum (entirely stealthful - domain only) to full network-level multicast. Most notably, stealth multicast allows a domain to deploy multicast support and realize concrete benefits without waiting for application, customer, or global routing support.

3 The MYDEKI Architecture

The MYDEKI (Multicast and You Don't Even Know It) is an architecture based on the stealth multicast model. The MYDEKI architecture governs the undefined areas of stealth multicast which include virtual group management, multicast transport, and state management. The MYDEKI architecture is targeted towards medium-size (tens to hundreds of clients) UDP[1] group-oriented applications employing separate unicasts. These applications may either operate with or without knowledge of MYDEKI. The MYDEKI architecture offers three modes of deployment, two stealth modes (differing by the location of the VGDM), and an application-assisted mode.

3.1 MYDEKI: Stealth Mode (FullStealth and Local)

In the stealth mode of MYDEKI, all applications are unaware of the presence that stealth multicast is being employed. The first contact that the architecture has with the packet occurs when the candidate packet is presented to the VGDM (see Figure 2).

Dispatch Mechanism: The intuition behind the MYDEKI parameters is to allow for predictable tuning by the network administrator. At their core, the MYDEKI parameters capture the potential benefits of waiting (receiving another packet that can be put into a virtual group) versus the effect that the queuing (delay, buffer size) is having on the behavior of the packet flow. For optimal performance, a packet should be kept as a group candidate so as long as only to add more egress points to the multicast group and hence increase the efficiency of the virtual group. Thus, MYDEKI includes triggers that reward close proximity of matching packets (likely to have another match) while still putting a firm cap on the maximum delay that can be experienced by a packet. The following parameters govern how MYDEKI manages the triggers for release for its virtual groups (see Figure 3):

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

[1] Although MYDEKI can be adapted for limited TCP support, such a topic is beyond the scope of this paper.

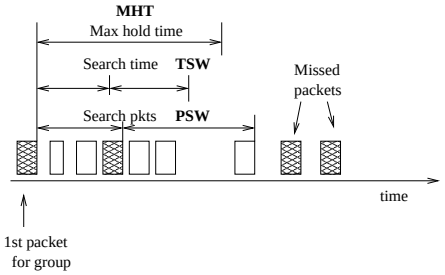


Fig. 3. MYDEKI - search settings

- *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 and places a maximum delay on the queuing time, regardless of additions to the virtual group.

Once a packet is triggered for release, it is given to the dispatch mechanism for dispatch to either the multicast modules or dispatch via standard unicast routing (no virtual group constructed). The MYDEKI group settings are discussed in more detail below:

- *MaxGS - Maximum Group Size:* This parameter forces a group to be dispatched when it passes a specific size.
- *MinGS - Minimum Group Size:* In order to be considered as a candidate for multicast, a group must meet a certain minimum membership level.

The 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 TSW reduces queuing time in the event of idle time on the link. 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 virtual group packets will be located close together.

The MHT setting allows one to controllably affect the delay impact of virtual group detection. The MHT places a worst case bound on the virtual group detection while the actual delay experienced will depend upon the underlying traffic patterns and the TSW/PSW settings. Hence, the PSW and TSW parameters limit the effective search for a given packet and limit the queuing delay of packets even further by emptying out non-growing virtual groups.

The notion of the total number of groups introduces a significant factor for setting of the dispatch triggers in MYDEKI, the maximum number of concurrent virtual groups (i.e. buffer size). Since the number of virtual groups and

the storage required for such groups and their packets cannot be considered infinite, the parameters should be set to minimize overflow conditions. Unlike traditional queue overflows where the packets are discarded, overflow packets are still forwarded onwards via unicast. Thus, overflow does not introduce catastrophic failure. However, once the capacity of virtual groups has been filled, no additional efficiency gains due to stealth multicast will occur until several of the existing virtual groups are dispatched.

Tree Construction & Multicast Transport: In order for a virtual group to be sent using multicast, it must first pass a minimum threshold ($MinGS$). Unlike standard IP multicast, the stealth multicast model introduces an overhead (transport information, state information) that must be offset by sufficient tree savings in order to offer an improvement in efficiency. The aspect of tree construction is encompasses how multicast packets are transported across the domain of the ISP. Since the destinations that make up the virtual group are not known a priori, we propose to use an encapsulation-based method for providing multicast transport across the domain. In short, an encapsulation-based approach includes the multicast tree inside the packet, thereby removing the need for multicast state along the routers in the multicast path. For MYDEKI, we have selected the DSMCast approach [7] as it was targeted towards multicast transport across a single domain. Although an encapsulation-based transport does introduce additional overhead due to the tree being encapsulated inside the packet, we believe the benefits of dynamic routing and resource management far outweigh the additional cost. If the minimum group size is set appropriately, the effects of the additional encapsulation overhead will be entirely offset by the multicast savings.

State Management: While the encapsulation-based transport allows for a stateless core to cross the domain, the unique portions of the packet (destination IP, destination port) must also be addressed. Furthermore, each unicast conversion must be appropriately associated with an egress point for the domain. Otherwise, it is extremely difficult for an egress point to accurately assess if it is responsible for converting a multicast packet for a specific destination.

Similar to how the encapsulation-based transport includes the tree information in the packet, the state information in MYDEKI is also bundled in the packet. In MYDEKI, each multicast packet includes three pieces of information for each client that is covered by the multicast packet, the egress point (who should convert the packet), the destination IP, and the destination port. Upon receiving a multicast packet, an edge (egress) router will inspect the packet to determine if it should appropriately convert/replicate the packet.

4 Simulation Studies

The simulations were developed using the ns-2 simulator and the GenMCast extension module for ns-2. The rationale behind our simulations was the following. In the network, Company X hosts an on-line gaming service with applications serving up to 120 clients. The application is hosted on a set of servers to users

outside of the initial domain of the company's ISP. The parameters for the network simulation are summarized in Table 1.

Parameter	**Setting**	**Parameter**	**Setting**
Maximum Groups	50	ISP Core Nodes	32
Maximum Hold Time (MHT)	10 ms	ISP Edge Nodes	16
Time Search Width (TSW)	2 ms	Total Sources	40
Packet Search Width (PSW)	100	Avg Clients	32
Min Group Size (MinGS)	2	Avg Packet Rate	50 ms
Max Group Size (MaxGS)	200	Avg Packet Size	500 bytes

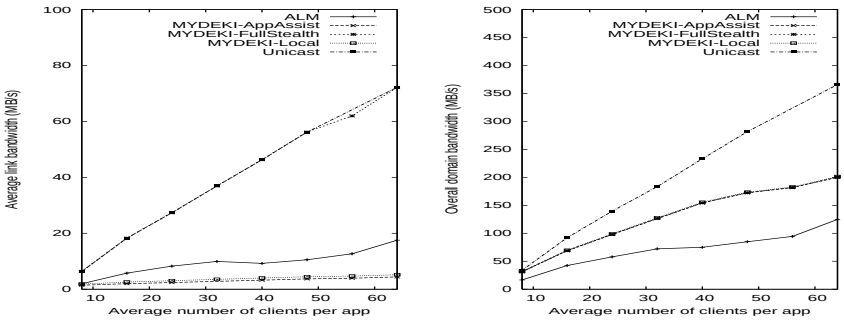
Table 1. MYDEKI settings

The primary purpose of the simulations is to evaluate the basic principles of the stealth multicast model (impact of queueing, predictability of control parameters, etc.). For our simulations, the MYDEKI model was evaluated according the following performance parameters:

- *Bandwidth utilization:* The bandwidth consumption of the server traffic on the uplink from Company X, the core of the domain, and the overall network were examined to determine the savings of the MYDEKI model.
- *End-user QoS:* The effects on end-user QoS and were examined to determine if the MYDEKI model was having a negligible or noticeable impact on the user QoS.

In our simulations, we compared the performance of five different distinct models under varying configurations:

- *Traditional Unicast:* In this model, no stealth multicasting is employed. This model is used as a base line for comparing the performance of the other two models.
- *MYDEKI-FullStealth:* 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 multicasting.
- *MYDEKI-Local:* In this model, the group detection module is placed directly at the edge router of the company. The traffic can be considered for stealth multicasting before going on the customer's uplink to the ISP.
- *MYDEKI-AppAssist:* In this model, the application actively participates by submitting state information to the VGDM at the edge of the ISP domain [5].
- *ALM:* A generic version of ALM was used that is based on End System Multicast [8].



(a) (b)

Fig. 4. Effect of average number of clients on (a) bandwidth - uplink (b) bandwidth - domain

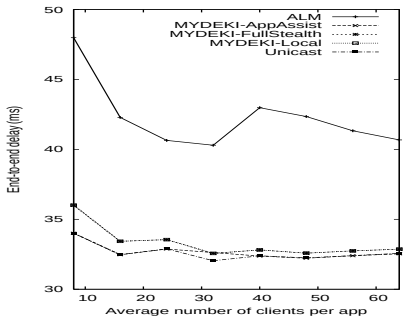
4.1 Effect of Client Subscriptions

The fundamental motivation for the stealth multicast model is to offer a significant bandwidth improvement in the core of the domain. Figure 4 plots the performance as the average number of clients per server application is varied from 8 to 64. 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. Whereas ALM offers better performance in the core of the network, it achieves such a balance by pushing the bandwidth consumption out to the edge links (uplink, client links) as evidenced by the Figure 4(a).

Most notably, the actual queuing delay of the VGDM is quite minimal as shown in Figure 5(a). Unlike ALM 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. Most important of all, stealth multicast can offer significant bandwidth improvements with zero modifications to the client or server applications ranging from 2x in the FullStealth case over the domain to over 10x for the uplink in the Local case.

4.2 Effect of MYDEKI - Packet Search Width

Figure 5(b) shows a negligble impact by the PSW on the performance of the various models. Due to the limited aggregation of the traffic (a single customer only), the packets frequently arrive in self-contained bursts rather than sporadically arriving due to aggregation with other flows. However, one can still discern the impact of an increased PSW in Figure 5(b) as an increase in PSW causes the VGDM to search a larger width and hence release the packet later from the queue. A inter-domain link employing a VGDM would experience a significantly increased impact of MYDEKI due to the TSW/PSW search widths being periodically reset rather than the burst of this scenario. Hence, the closer



(a) (b)

Fig. 5. Effect on end-to-end queuing delay of (a) average number of clients and (b) packet search width

the VGDM can be placed to the customer, the less that MHT must be used to cap the maximum delay and the more that TSW/PSW can be relied upon (rewarding proximity).

4.3 Effect of Aggregation in Client Packets

While the ideal case would be to only monitor traffic that will be part of a virtual group, such will most probably not be the case in practice. Hence, Figure 6 plots the performance of the various models as the probability of variance of client packets is varied from 0 to 1.0. In short, the probability of variance captures the chance that a given packet in a server transmission will be unique from all other packets (distinct checksum). As the variance is increased, more and more of the packets are sent out as unique packets that will become a 1-packet virtual group. The earlier plots of the number of clients had a variance of zero since variance represents intra-group distinct packets that neither AppAssist nor ALM models could handle (transmission to a partial subset of clients).

5 Related Work & Summary

While several works have base similarities with stealth multicast, the work in this paper is unique in that it is the first work to dynamically aggregate redundant payloads for multicast transmissions. The closest work is in [6] where packet-level caching was applied. Other works include ConCast [9] and GatherCast [10]. However, stealth multicast fundamentally differs from these two approaches in that ConCast provides a many to one multicast while GatherCast aggregates multiple small packets into a larger packet (still preserving the content of the packets).

The stealth multicast model offers a technically feasible approach for solving the issue of economic incentive for ISPs to deploy multicasting. Although the

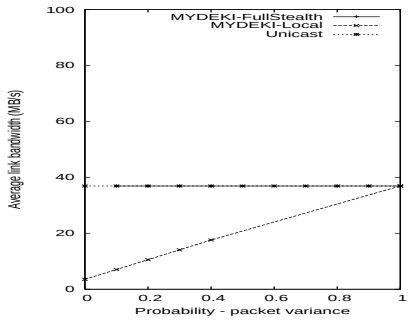


Fig. 6. Effect of packet variance on bandwidth - uplink

stealth multicast model does not ensure that multicast will be deployed multicast in a global sense, it addresses many of the fundamental issues that hinder multicast on even a domain-wise level. Furthermore, we believe the stealth multicast model offers a unique approach to multicast and provides an excellent platform for future research.

References

1. Almeroth, K.: A long-term analysis of growth and usage patterns in the multicast backbone. In: Proc. of IEEE INFOCOM'2000. (2000)
2. Beverly, R., K. Claffy: Wide-Area IP Multicast Traffic Characterization. IEEE Network (2003)
3. Diot, C., Levine, B., Lyles, B., Kassem, H., Balensiefen, D.: Deployment issues for IP multicast service and architecture. IEEE Network (2000) 78–89
4. El-Sayed, A., Roca, V., Mathy, L.: A survey of alternative group communication services. IEEE Network (2003)
5. Striegel, A.: Dynamically encapsulated trees for stealth multicast with mydeki. Technical Report TR-04-11, Univ. of Notre Dame Comp. Sci. and Engr. (2004)
6. Spring, N.T., Wetherall, D.: A protocol independent technique for eliminating redundant network traffic. In: Proc. of the 2000 ACM SIGCOMM Conference, Stockholm, Sweden (2000)
7. Striegel, A., Manimaran, G.: A scalable protocol for member join/leave in DiffServ multicast. In: Proc. of Local Computer Networks (LCN), Tampa, Florida (2001)
8. 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) To Appear.
9. Calvert, K.L., Griffioen, J., Mullins, B., Sehgal, A., Wen, S.: Concast: Design and implementation of an active network service. IEEE Journal on Selected Area in Communications (JSAC) (2001)
10. 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)