Adaptive LAN-to-Host Multicast:

Optimizing End System Multicast via LAN Multicast Integration

David I. Murray
SCS Senior Honors Thesis
Carnegie Mellon University

Spring 2006

ABSTRACT

In this paper, I report on the findings of designing and implementing a significant optimization to Overlay Multicast, also known as End System Multicast (ESM). The current system uses a peer-to-peer approach to propagate data through hosts in a waterfall-like tree structure, leaving a bandwidth cost of O(1) to the source. Although peer-to-peer systems like ESM have been revolutionary in their bandwidth efficiency and scalability across networks, a problem arises when we look at the traffic on a single Local Area Network (LAN)—the overlay tree structure can only guarantee that traffic is linear to the number of users tuned in. In the same situation, another known protocol, IP Multicast, can guarantee a constant level of traffic. The problem with IP Multicast, however, is that it cannot be guaranteed to function across LANs, because many routers do not support the protocol, hence the need for ESM. The purpose of my research is to examine and discover how to use the benefits of both protocols in order to combine them and achieve significant bandwidth savings for all hosts. The resulting system that I designed and implemented uses a channel-independent waterfall structure in which the elements of the waterfall are LANs instead of individual machines. Each of these LANs uses LAN Multicast within to propagate data, and, because the system is channel-independent, multiple channels of content can be delivered. This combination has ensured a level of bandwidth and traffic efficiency that is optimal in all situations: an O(1) bandwidth cost to the source, an O(1) level of traffic on the LAN, and the potential to broadcast hundreds of video and audio content channels to millions of users at virtually no cost to the sources or participating hosts. The key findings from evaluating this implementation verify and conclude that this optimization is, indeed, achievable—moreover, it is what this new system has already achieved.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols – *applications, routing protocols.*

General Terms

Performance, Design, Reliability, Verification.

Keywords

Networks, Peer-to-Peer, IP Multicast, LAN Multicast, Overlay Multicast, End System Multicast.

1. INTRODUCTION

Peer-to-peer (p2p) technology is arguably one of the most pervasive topics in networks research today. Its creation spawned from efforts to distribute content to large numbers of users in a way that is both efficient and scalable. One of the earliest solutions to this problem was IP Multicast, "a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of recipients." [3] Although viable, IP Multicast has failed to pan out on a global scale, as it requires routers to implement IP Multicast-handling features, which many routers do not support because of the additional processing required. As a result, researchers investigated alternatives that can provide the same degree of scalability and efficiency as IP Multicast but leave no dependence on individual router configuration. The result was peer-to-peer technology.

Popular p2p protocols like BitTorrent [5] and Gnutella [6] are in heavy use today, both commercially and otherwise, and the rising approval for mainstream use of such protocols is on the rise. There is, however, a great deal of work left to be done, as specific application demands have brought the need for tailored protocols. In regard to Internet video and audio broadcasting, protocols like BitTorrent and Gnutella do not offer speed and bandwidth guarantees, hence research has been underway to find the right peer-to-peer solution for Internet broadcast and other high-bandwidth, speed-critical applications. It is from this research that Overlay Multicast, or End System Multicast (ESM), found its beginnings.

The idea behind ESM is that, instead of relying on the network layer, end-systems can organize themselves at the application layer in such a way that they distribute the content amongst each other. Using this approach, the end-systems are not bound to router configuration topology constraints, and they can be organized in such a way that is optimal for a given application. This could be in the form of a waterfall-like tree, a linear ordering, or an arbitrary structure that best suits the application at hand.

Because most Internet broadcast applications are concerned with distribution of content to as many hosts as possible as quickly as possible, a common structure of choice is a waterfall-like tree, in which the source of the data forwards packets to one or more hosts, those of which forward to other hosts, and so on (Fig. 1). The result is that n hosts receive the data, where n is any value, and the source of the broadcast uses O(1) bandwidth instead of O(n) bandwidth as required in the non-multicast server-client broadcast case.

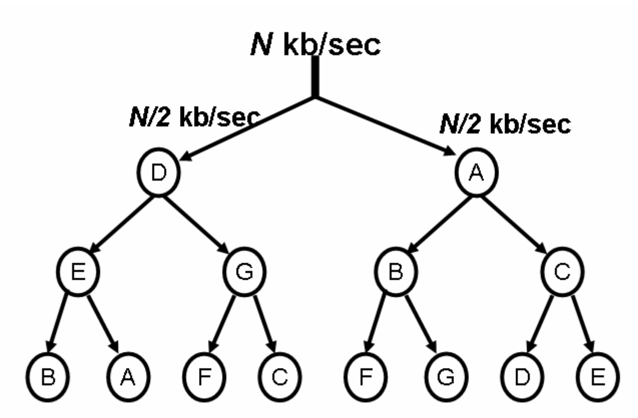

Figure 1. A single overlay distribution tree

The Carnegie Mellon End System Multicast Group has been among the forerunners in researching this technology with work having been underway for the past few years in developing a peer-to-peer overlay system for Internet broadcasting. The goal of their research is to meet "the vision of enabling live video broadcast as a common Internet utility in a manner that any publisher can broadcast content to any set of receivers" [1]. To achieve this goal, they designed a system that utilizes a waterfall-like tree structure to broadcast both high- and low-quality broadcasts to thousands of users. This structure originally began as a one-directional waterfall (Fig. 1), however it has developed into a multiple overlay distribution tree (Fig. 2), in which one main tree is used, but redundant links are provided so that bandwidth losses can be quickly repaired.

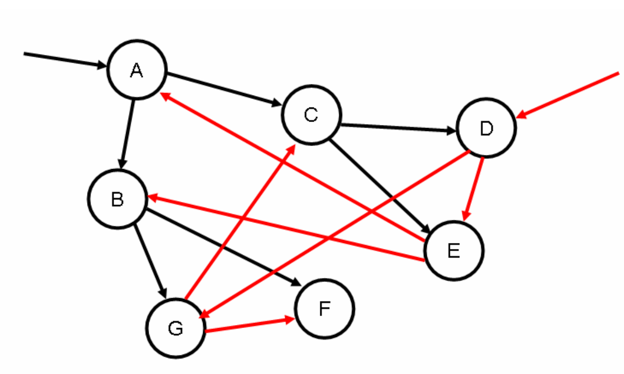

Figure 2. A multiple overlay distribution tree

2. PROBLEM

As Overlay Multicast systems have become more robust, research has expanded to encompass related pressing issues like security, resilience, deployment, and scalability. In regard to the latter, let us examine what happens at the level of a single LAN when ESM is used as compared to IP Multicast.

2.1 LAN End System Multicast

Let us consider the best case, namely that the users within a single LAN that are tuned into a broadcast are closest to each other in the distribution tree than everyone else tuned into the broadcast. If n users are tuned into the broadcast, then, we could have a variety of reasonable topologies for the hosts in the LAN:

- Each host has zero or two children, similar to Fig. 1
- Each host has exactly one child, forming a linked list-like structure
- One host has all of the other hosts as children
- Each host has some variable number of children

Regardless of the case, there must be $n – 1$ distinct "connections" (we defined a connection to be the link between a child host and its parent host upon which broadcast data is transferred); this comes from the fact that a tree with n nodes has $n – 1$ edges. Since each of these $n – 1$ streams contains the same data, there are $n – 1$ duplicates of a given data packet being propagated on the LAN at any given time. As such, the level of broadcast traffic on the LAN is $O(n)$.

2.2 LAN IP Multicast (a.k.a. LAN Multicast)

Without loss of generality, let us assume that the data source is on this LAN.[1] There is only 1 case here: a source distributes one stream of data to an IP Multicast address on the LAN, and all hosts that are subscribed to that multicast group will receive this data. Because there is only one stream, the level of broadcast traffic on the LAN is O(1).

2.3 LAN ESM vs. LAN Multicast

In conclusion, we observe that ESM generates $O(n)$ broadcast traffic on the LAN whereas LAN Multicast generates O(1) broadcast traffic. This indicates that it would be preferable to use IP Multicast to propagate data instead of ESM. However, if we consider applying this rule across LANs, we run into the same problem that was described earlier, namely that we are dependent on routers which may or may not be configured to support IP Multicast.

In the ideal world, we would utilize the efficiency of LAN Multicast for broadcasting within a LAN, but we would utilize the pervasive deployability of ESM to broadcast between networks, as this would allow us to ignore routers configurations. In addition, to prepare for the future of Internet broadcasting, the capability to support multiple channels of content is necessary—although this area has not been fully explored by previous research, I wished to address it and integrate the solution into my system.

If we harnessed the benefits of End System Multicast and LAN Multicast, combining them in a way that efficiently and robustly supports multiple channels of content, we can achieve an O(1) bandwidth cost to data sources, an O(1) level of traffic on every LAN, and the potential to broadcast multiple channels of video and audio to theoretically millions of users at virtually no cost to the source or the participating hosts. This is the goal I set out to achieve.

3. RELATED WORK

A broader variant of this problem was addressed by researchers at the University of Michigan in 2002. [3] Instead of considering the

[1] We can generalize to the case that it exists outside the LAN, because there is at least one host on the LAN that servers as the point of entry of data from outside.

case of individual LANs, this paper addressed the issue of IP Multicast "islands of network domains under single administrative control," which may be a single LAN or a collection of LANs united by some topology. The system they developed was a hybrid protocol, Host Multicast Tree Protocol (HMTP), which uses IP Multicast where available and Overlay Multicast to connect IP Multicast-enabled islands.

The work from this paper left three critical unresolved issues that I wish to address:

- IP Multicast-enabled islands of sizes and topologies that invisible to the HTMP protocol, creating potential performance bottlenecks
- Leader bottlenecks resulting in potential infrastructure collapse
- Lack of support for multiple channels of content

A. IP Multicast-enabled Islands vs. LANs

In essence, HTMP was intended to serve as a more deployable alternative to applications like MBone, which require manual configuration to connect IP Multicast enabled "islands" into static overlay networks. The problem that they addressed is similar to that which I address in this paper, in that the goal of the work was to find an effective means to combine Overlay Multicast with IP Multicast. A significant difference, however, was in their choice to consider entire IP Multicast domains. In this case, no assumptions can be made about the topology of a given island; hence the propagation delay for entire island saturation is unknown. For instance, smaller topologies and individual LANs can guarantee quick delivery, whereas larger IP Multicast islands, like country-divided corporate networks, may take more time to fully propagate. Either way, this information is invisible to the HTMP implementation, so it cannot be utilized.

This is clearly a limitation when considering the design for an End System Multicast framework to integrate with IP Multicast. Because no reasonable propagation delay assumptions can be made on a given IP Multicast-enabled island, it is difficult to design a time-critical reliability structure within the island to support End System Multicast. No island topology information is known; hence the most appropriate hosts to serve as End System Multicast delegates[2] for a given island cannot be determined without significant additional overhead. The result is that they cannot guarantee a reasonably time-critical propagation of data.[3]

If we consider, instead, individual LANs[4] in all contexts as opposed to IP Multicast-enabled islands of various topologies and sizes, we know by definition that the domains we are working with share a common communication line at the link layer. Routers are completely out of the picture. Hence, we can use this information in the design of our protocol to take advantage of the high-speed guarantee that an Ethernet LAN can provide.

B. Leader Bottlenecks

The HTMP protocol elects a Designed Member (DM) on each IP Multicast island to have the following responsibilities:

- Decapsulate data packets that are tunneled from outside the IP Multicast island
- Propagate data from other islands to all members of the given island
- Maintain a list of IP Multicast groups of interest and listen to them
- Establish and maintain tunnels to other IP Multicast islands
- Tear down island tunnels to islands that are not of interest

We observe, hence, that one host handles both data-distribution leadership and control-plane leadership for the entire IP Multicast islands. When the DM terminates, the result is the collapse of data and control on the IP Multicast island and the disintegration of data flow for all hosts on nearby islands that receive data from this one. It is unclear from the paper how DM election takes place or how long it takes, and whether the election is done upon abnormal DM termination, but the fact that the DM is responsible for data propagation ensures that all members of the island will not be receiving data while the election is underway.

A significant disadvantage to centralizing data distribution and control-plane leadership to one host is that the only data that can be propagated through the island must be done via one host. This is certainly a potential bottleneck; it is for this reason that the need for multiple DMs is expressed in the "Summary and Future Work" portion of the paper. [3]

C. Multiple Channel Support

Another consequence of the dependency on a single host for data distribution is that the number of channels of data that can flow in is limited to how many channels this host can handle. Hence, we cannot rely on the scalability of this system to support multiple applications or multiple channels of video/audio broadcasts. It might be possible to support two or three high-quality streams of broadcast data via one host, but if we want to expand to a higher order of magnitude to support hundreds or thousands of streams, this bottleneck will ruin our efforts. To some degree, we are back to the problem we started with, in that we cannot scale our system to support massive levels of content distribution.

D. Other limitations

The following limitations of HTMP have also been provided by the paper which discusses it:

- "Applications must inform the Host Multicast agent about their interest in any particular multicast group. This can be done by a function call to the Host Multicast library."
- "Applications and the agent must turn on multicast loop-back. This is to ensure that applications and

[2] We define a "delegate" to mean an end host which receives data via Overlay Multicast from outside the island and transmits within the island via IP Multicast and/or outside the island via Overlay Multicast.

[3] Because all of these systems are best-effort, I use the term "guarantee" to mean "achieve as close to a guarantee as is possible in a best effort environment".

[4] As defined by the use of Time-to-live = 1.

agent[s] on the same host can receive IP Multicast packets from each other."

- "Since decapsulated packets are sent by the agent, and not by the original data source, applications may no longer rely on source IP address[es] on data packets to identify the original data source. Even if the agent uses raw socket[s] to change [a given] source IP back to the original one, it still does not work in some cases." [3]

Hence, it is clear that more work on integrating End System Multicast with IP Multicast is necessary to achieve new breakthroughs in efficiency, reliability, and scalability. A solution is needed that can provide the robust and resilient ability to broadcast many channels of data, free from bottlenecks and information alteration. This is the mindset upon which I designed what I call Adaptive LAN-to-Host Multicast (ALHM).

4. DESIGN

4.1 Requirements

As stated above, efficiency, reliability, and scalability are all requirements of ALHM. In addition, the ALHM implementation has the following requirements. It needs to:

- Contain End System Multicast functionality between LANs
- Contain LAN Multicast functionality within each LAN
- Be able to adapt data propagation between End System Multicast and LAN Multicast where appropriate
- Ensure consistent data flow and robustness by swift replacement of important hosts that terminate
- Be resilient to normal and abnormal host joins/leaves
- Be capable of handling magnitudes of different content channels without persistent performance bottlenecks

Finally, although not a requirement, my code needs to be modular. Since I am integrating ALHM into the current CMU End System Multicast proxy implementation, using a modular code-base will allow for the current protocol to remain intact while still providing inter-protocol reliability.

4.2 Solution Abstraction

On an abstract level, ESM is based on an overlay distribution tree in which the nodes of the tree are individual host machines. Conceptually, I am replacing individual hosts with LANs (Fig. 3). This is not to indicate anything about which hosts on a given LAN are points of entry for End System Multicast, but conceptually, this image serves as the foundation of my design.

This abstraction has implications on both the data-plane and the control plane of my design. In regard to the data-plane, traffic is propagated to individual system(s) on each LAN, and that data is then assumed to be propagated to all members of the LAN (this is done by LAN Multicast). In regard to the control-plane, LAN-level control is essentially modularized from the main ESM system; responsibility for sending correct join/leave information to the main ESM system is now up to each LAN (i.e. representatives of that LAN) instead of each host.

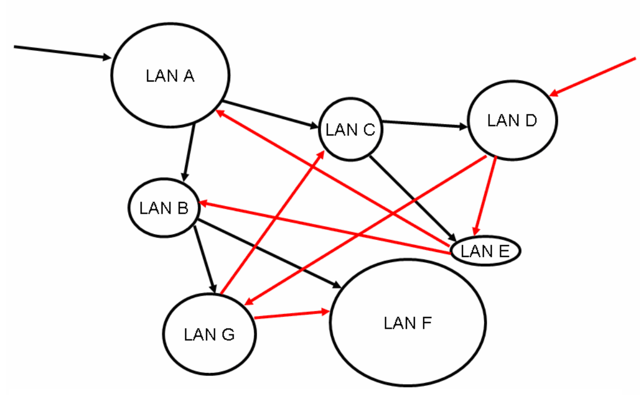

Figure 3. The ALHM data distribution abstraction

The final abstraction that I determined was necessary in my design was that LAN data distribution topology control information should be communicated independent of broadcast data channels. This is to ensure that the control protocol does not pose any limitations to the data flow, as was in the cast in HTMP.

4.3 Solution Elements

In order to make these concepts a reality, I determined two separate types of leaders for each LAN: the control-plane leader, which I call the LAN Facilitator, and the data-plane leader(s), which I call the Channel Forwarders. All members of the LAN that are not Channel Forwarders are called Channel Participants. When a host joins ESM, it is designated to be one (or more) of these elements.[5]

A. LAN Facilitator

The LAN Facilitator serves as the center of protocol control for a given LAN; it communicates to the hosts tuned in on the LAN via a reserved multicast group for ESM. Each LAN has a Facilitator, and this Facilitator does not communicate control messages outside the LAN.[6] Responsibilities of the LAN Facilitator include:

- Handling when hosts on the LAN join/leave a channel
- Handling when hosts on the LAN join/leave the system
- Designates and notify a backup host to replace in case it goes down in the future
- Designate which host(s) are Forwarders for each given data channel
- Reassign Forwarders upon Forwarder channel leaves
- Timeout hosts that have not been heard from in a while (i.e. 1-2 seconds)
- Send keep-alive messages on the control-plane multicast group

---

[5] How this is done is described in *4.4 Control Mechanisms.*

[6] This is done by the use of LAN Multicast, which, as a reminder, we define to be IP Multicast with time-to-live = 1.

Table 1. The three host-elements of ALHM

Type	LAN-Unique	Channel-Unique	Requires swift replacement
Facilitator	Yes	(channel independent)	Yes
Forwarder	No	No[7]	Yes
Participant	No	No	No

B. Channel Forwarder

A Forwarder serves as one of the hosts that tunes into the global ESM tree for a given channel of data and forwards that data to all members of the LAN via a unique multicast address. This address is determined via a hash function that takes a channel ID and returns a multicast address. Responsibilities of a Channel Forwarder include:

- Connecting the global ESM tree, where it is assigned global parents/children for data distribution
- Forwarding data received to members of the LAN via a channel-unique multicast address
- Send keep-alive messages to the LAN Facilitator
- Notifying the LAN Facilitator when it wishes to leave a channel; it then waits until the notification has been acknowledged to leave the channel and leave the global ESM tree for the channel

C. Channel Participant

A Channel Participant is simply a host that is not a Channel Forwarder. To receive data, the Participant subscribes to the multicast address corresponding to the given channel's ID, as determined via the hash function that is used by the Forwarder. Responsibilities of a Channel Participant include:

- Listen to a channel-unique multicast address for data
- Send keep-alive messages to the LAN Facilitator

4.4 Control Mechanisms

End System Multicast works at the application layer. Each host runs a proxy application which connects to ESM. The proxy forwards all broadcast data to the host's loop-back address, and the media player then tuned into the loop-back address to play it. Based on these constraints, the constraints of LAN Multicast, the need for modularity, and the requirements previously described, I determined that for ALHM to work, there are 6 mechanism which must be in place:

1) Connection establishment
2) LAN member discovery
3) Channel member discovery
4) Data stream forwarding
5) Facilitator election and maintenance
6) ESM-to-LAN Multicast adaptability

A. Connection Establishment

Upon startup, the ESM proxy application obtains a unique 6-byte identifier.[8] It then sends a JOIN message to the ESM reserved control LAN Multicast address. If it receives no response within half a second, it designates itself to be the LAN Facilitator; otherwise, the LAN Facilitator will reply to the JOIN with a JOIN_RESPONSE.

In either case, the LAN connection is established within half a second, and the result is that a LAN Facilitator will always be in place if anyone on the LAN is using ESM. If the JOIN_RESPONSE packet is lost and it is later discovered that another host thinks that it is the facilitator, a FACILITATOR-_OVERTHROW packet is sent, upon which this facilitator replies with another FACILITATOR_OVERTHROW packet. The host with the lowest id remains the Facilitator while the other relinquishes.

In all cases, the Facilitator will always designate and notify a host to be its backup incase it terminates. This backup will then takeover whenever the Facilitator terminates (details on this are in section *E. Facilitator Election and Maintenance*).

B. LAN Member Discovery

The LAN Facilitator expects that hosts that join ESM will send a JOIN message to the ESM control multicast group. However, in the case that a host proxy continues after being suspended for a period of time, or in the unlikely event that the JOIN packet is lost on the LAN, the LAN Facilitator may receive a packet from an unknown host on the LAN. When this occurs, the Facilitator sends a REQUEST_REFRESH packet, which instructs the host to inform the Facilitator which channels it is tuned into and for which channels it thinks it is the Forwarder. The facilitator then incorporates this information and resolves and topology discrepancies that result.

C. Channel Member Discovery

When a host wishes to join a channel, it will initially connect to the hashed multicast group for the channel under the assumption that it will be assigned to be a Participant for the channel. It then sends a channel-specific JOIN request to the LAN Facilitator. The Facilitator will reply with an assignment; if there is no Forwarder for the channel, it will be assigned as the Forwarder; otherwise it will be assigned as the Participant. In the case that it is assigned to

[7] Although there can be more than one LAN Forwarder for a given channel of data, only one is necessary. A second one may be used for redundancy in the case that the first fails, but only one host needs to stream data to the LAN multicast address. As such, only one host is used in all broadcasts in ALHM by default.

[8] This is currently done by contacting a host called the ESM Decision Element (DE) which is part of the CMU ESM system. Every time a user starts a broadcast, a message is sent to the central ESM website which then starts this DE application. In the near future, unique IDs will be channel-independent to support the broadcasting of multiple data channels.

be Forwarder, it will join the global ESM tree.[9] All data packets received by this host will now be forwarded to the LAN via LAN Multicast.

D. Data Stream Forwarding

Data forwarding is something that is completely maintained by the current ESM system. The only modification that is made to this system with the integration of LAN Multicast is that a Forwarder must add the channel multicast group to its list of children so that data can be propagated to the LAN via LAN Multicast. Participants tuned into this multicast address then received the data and pass the unmodified packets directly to the ESM data handling code.

E. Facilitator Election and Maintenance

Three cases need to be addressed here:

1) The Facilitator terminates normally with Backup up
2) The Facilitator terminates abnormally with Backup up
3) The Facilitator *and* its Backup terminate abnormally

If the Facilitator terminates normally and there are no other hosts on the LAN that it is aware of, there is nothing left to do. If there are other hosts, however, the Facilitator notifies its backup of a request to terminate. The backup responds to this request by becoming the facilitator and sending a REQUEST_REFRESH packet on the ESM control multicast group, requesting all hosts on the LAN to inform it of which channels it thinks it is participating in and which it thinks it is forwarding.

If the Facilitator terminates abnormally, it will stop sending KEEP_ALIVE packets. The Backup will be the first to notice, upon which it will broadcast a FACILITATOR_TAKEOVER packet to the ESM control multicast group and begin transitioning into the Facilitator. If a host receives a FACILITATOR-_TAKEOVER packet, it responds with a REFRESH packet containing its channel participation information. After 1 second to allow the REFRESH packets to come in, the Backup will fully transition into the Facilitator, upon which it will broadcast a FACILITATOR_OVERTHROW packet to ensure that no other hosts think they are going to become the Facilitator. After this, it will consider the data received in all REFRESH packets, and repair any channel participation and forwarding conflicts that developed during the transition.

In the uncommon case that both the Facilitator and its Backup terminate abnormally at about the same time, Facilitator election takes place. After two seconds of not receiving a KEEP_ALIVE or another response from the Facilitator, each host will determine that the Facilitator has terminated. In this case, each host will send a FACILITATOR_ELECTION packet to the ESM control multicast group and wait for 1 second.

If it has not received a FACILITATOR_ELECTION packet from someone else with a lower ID, it will attempt to begin the transition into facilitator in the same way the Backup does, first sending a FACILITATOR-_TAKEOVER packet, then waiting a second for incoming REFRESH packets, then finally taking over by sending a FACILITATOR_OVERTHROW packet.

The result of these Facilitator maintenance mechanisms is that the only case that may take more than 1-2 seconds of time is the case in which both the Facilitator and the Backup terminate abnormally at nearly the same time and Facilitator election becomes necessary. Even this case should take no more than 3 seconds to complete.

This is one of the contexts in which we benefit from decoupling data management with control-plane management. The fact that these Facilitator processes are independent of what occurs on the data-channels means that end-users will not notice any change in video/audio transmission unless the Forwarder for the given channel terminates abnormally during these 3 seconds (in the case of normal termination, the Forwarder would wait until a new Facilitator has been elected and acknowledged its leave request).

F. ESM-to-LAN Multicast Adaptation

The final control mechanism is the adaptation on the LAN between using ESM and using LAN Multicast. In my implementation of ALHM, ESM is exclusively used if only one host is using ESM on a given LAN. Although this host determined itself to be Facilitator, it does not forward any data or do any other control management because no other hosts are there. When a second host joins, however, the control mechanisms come into effect, and either the Facilitator or the other host will be designated as the Forwarder for any channels to which they are tuned into. This is what I mean by ESM-to-LAN Multicast adaptation.

Because LAN Multicast comes into effect when 2 hosts are using ESM on the LAN, the threshold number of users upon which a LAN transitions from ESM to LAN Multicast is 2. As I will discuss at the end of this paper, it may be beneficial in certain situations to have a higher threshold, especially in a possible future scenario in which hundreds of channels or more exist on a LAN but each channel only has a few users tuned in.

5. IMPLEMENTATION

The preceding control mechanisms are all which needed to be put into place to make ALHM into an implemented reality. To integrate with the current CMU End System Multicast proxy application, it was necessary for me to implement these mechanisms with as little change as possible to current ESM code.

As such, I took the extreme modular approach of first writing my own standalone application which uses all of the above elements and mechanisms but simulates joining ESM and propagating data.[10] Hence, I created the infrastructure for LAN Multicast and designed skeletons for the potions of code which I planned to link with the main ESM system. Upon testing this standalone application thoroughly, I began to integrate it into the main ESM system.

[9] This is done by contains the DE for the broadcast channel, which will then assign the node data parent(s) and/or children(s).

[10] All of code was originally written in C, however it has since been adapted to C++. This is for the sake of general modularity and compatibility with CMU ESM which is also written mainly in C++.

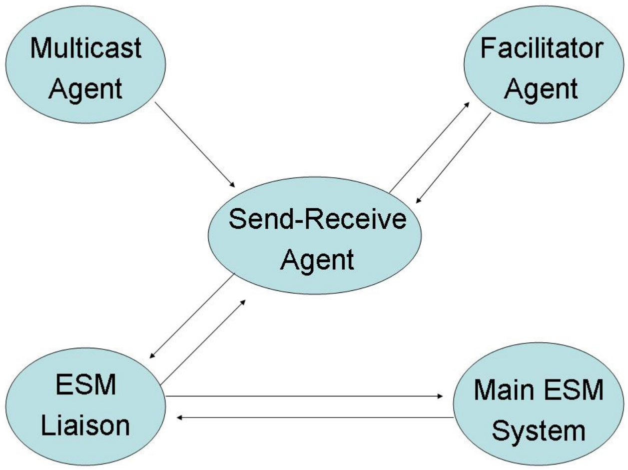

Figure 4. The ALHM modularized agent class structure

5.1 Standalone Implementation

With modularity as my motivation, I developed four separate agent classes to do processing for different levels of ALHM (Fig. 4). They are as follows:

- LAN Multicast Agent
- LAN Multicast Send-Receive Agent
- LAN Multicast Facilitator Agent
- LAN Multicast ESM Liaison

A. LAN Multicast Agent

This class served as an entry point for LAN Multicast functionality from the main ESM system. As such, the functions of this class serve to complete application-level requests, such as joining or leaving a channel, joining or leaving ESM, and setting and handling file descriptors to listen for and handle ALHM packets.

B. LAN Multicast Send-Receive Agent

The send-receive agent is used by the Multicast agent to handle general ALHM packet handling and transmitting. This code is mainly called by the LAN Multicast Agent. In addition to packet send and receive functions, this class maintains a timer to handle keep-alive transmission at the rate of once every 500ms. It also handles facilitator election and maintenance in the case that the given host is not the Facilitator for the given LAN.

C. LAN Multicast Facilitator Agent

This class contains all of the Facilitator-specific functionality that only the LAN Facilitator would use. This includes facilitator-specific transmission and handling, and it also includes channel member maintenance. This is the level at which LAN members are assigned to be either Forwarders or Participants for the channels to which they are tuned in. This code also has a periodic check that expires any hosts that have not been heard from in more than two seconds. When this expiration occurs, the Facilitator iterates through its list of host and channel information and appoints any new Forwarders where appropriate.

D. LAN Multicast ESM Liaison

This is the code that integrates with the control mechanisms of the main ESM system to ensure that the LAN Multicast control plane remains consistent with what is happening at the End System Multicast level. It is at this level that Channel Forwarders join the main ESM tree to receive and transmit data, and where Channel Participants subscribe to the hashed multicast group for a given channel to receive data. This is also the class through which data packets received from LAN Multicast are sent to the main ESM system for processing.

5.2 ESM Integration

When these four classes were fully functional and tested thoroughly in my simulated ESM environment, I began the process of integrating these elements into the actual main ESM system. Three modifications on ESM were necessary to make this happen:

- Data deployment integration
- ESM packet modifications
- Decision Element (DE) modifications

A. Data Deployment Integration

The functionality of the LAN Multicast ESM Liaison agent at this point was a skeleton that needed to be filled with function calls to the actual ESM system. There are two agents in the ESM system which needed to be reached to achieve success in actual data deployment: the ESM Gossip Agent and the ESM Control Agent. Each works in conjunction with one another to handle ESM control-plane communication. It is through the main ESM Gossip Agent and Control Agent objects that the proxy application can join the global ESM tree. This is done by contacting the ESM Decision Element (DE) which assigns parents and children.

I thus added links to the Gossip Agent and Control Agent within my ESM Liaison code, and call their ESM tree join/leave functions whenever a LAN Forwarder requests to join the main ESM global tree for a given channel.[11] In addition, I added a hook to the ESM Data Agent within the ALHM ESM Liaison that allowed for data packets that were received to be passed into the main ESM system for processing and propagation to the loop-back address for video player playback.

B. ESM Packet Modifications

With the integration of LAN Multicast Channel Participants comes the concept of hosts that are connected to the main ESM system but do not participate in the global ESM channel tree, because they receive their data from the Channel Forwarder. This concept was not one that existed in the current system, because the meaning of “Join ESM” was coupled with the idea of “Put me into the ESM tree”.

---

[11] It should be noted here that the current ESM application proxy is only capable of tuning into one channel at a time. This will change in the future, and the ALHM code is fully compatible with the use of multiple channels by design.

To decouple the two concepts, it was necessary to create two new types of ESM packets:

1) A GET_ESM_ID packet

2) A LAN_PARTICIPANT_KEEP_ALIVE packet

The first packet is used at the initial connection to ESM as a means of getting a unique identifier without connecting to the main ESM tree. The second is a special type of message that Channel Participants send to the DE to provide bandwidth information and other quality of service information. By sending this message instead of the normal DE keep-alive messages, hosts are now able to communicate to the DE without being forced to enter the main ESM tree structure. The result is that hosts can now join the ESM main system without having to join the main ESM channel tree.

C. Decision Element (DE) Modifications

The previous modification also required modifications at the DE to handle these new packets. Upon the receipt of a GET_ESM_ID packet, the DE now needs to obtain an ID but not put this host into the main ESM tree. In addition, when it receives LAN-_PARTICIPANT_KEEP_ALIVE packets, it needs to record bandwidth information but not incorporate the host into the main ESM tree.

In future updates to ESM, it is possible that Channel Participants will propagate data within the Global ESM tree while continuing to receive data on the LAN. These DE modifications make this update possible, as the only modification required would be to assign children to Channel Participants, and for the Channel Participants to inform the DE when they leave the channel (right now, it is not necessary to tell the DE anything, because the Channel Participants do everything they need to via contacting the LAN Facilitator).

6. ANALYSIS

In some regards, analyzing this system is synonymous to building it and verifying that it functions properly; if we can verify that the ALHM implementation works the way it is described and sends and receives packets as has been described, then we know by definition of our protocols that our goals are achieved—an O(1) level of traffic is propagated on each LAN for a given channel, and the data sources only require O(1) bandwidth to propagate their broadcasts. Nevertheless, we justify that the bandwidth levels in the real world system correspond to this hypothesis with data.

6.1 Methodology

6.1.1 Normal Case Bandwidth Measurement

The trivial scenario to verify is that a host can successfully tune into a LAN Multicast address and receive data near or at 100% of the bandwidth at which the data is being transmitted. To do this, one needs only start a broadcast, after which he must start two hosts on a separate LAN. One of these hosts will be the LAN Facilitator and the other will tune into the LAN Multicast address for data. We can then measure bandwidth receipt levels of the host tuned into the multicast address for data and verify that they are at or very near to the data transmission rate.[12] As a reasonably high rate of broadcast on today's systems, I chose 400kpbs as my transmission rate to test upon; all subsequent tests were done using this rate as well.

6.1.2 Risk Analysis Methodology

In addition to analyzing bandwidth levels in the normal ALHM data transmission case, it is also worthwhile to focus on cases in which bandwidth levels are at risk in ALHM. In terms of bandwidth risk analysis, there are a few observable contexts in which it is valuable for us to examine host bandwidth levels:

- The LAN Facilitator terminates abnormally
- The Channel Forwarder and LAN Facilitator terminate abnormally
- The LAN Facilitator and its Backup terminate abnormally, resulting in Facilitator Election

A. Abnormal LAN Facilitator Termination

Because the Facilitator is not responsible for data transmission to the LAN unless it is a Forwarder for a channel, LAN Facilitator termination will only have an effect on data flow if the Facilitator is also the Forwarder for the channel of concern. Hence, if we terminate the Facilitator abnormally but allow the Forwarder to continue transmission, no data loss should be noticed by the hosts on the LAN.

To examine this case, I started 3 hosts in them same LAN with ESM, having my second host be the broadcast source, sending data at a rate of 400kbps. The first host to join the system became the LAN Facilitator. When the second host joined, the Facilitator designated it to be the Channel Forwarder for the LAN because it is the source of the data flow for the entire broadcast. I then joined a third host and let this system stabilize by waiting 5 seconds.

After 5 seconds, I terminated the Facilitator. During the 40 second period encompassing this termination, I examined the bandwidth levels of the third host that was receiving data from LAN Multicast.

B. Abnormal Forwarder and Facilitator Termination

Although the data-plane is decoupled from the control-plane in the ALHM system, it is possible for the LAN Facilitator to also be the Channel Forwarder for one or more channels of data. Hence, if we terminate this host, we are terminating both the flow of data and the flow of control on the LAN. Fortunately, Facilitators designate backups which immediately replace the Forwarder one it is recognized to be down. To result is that a swift recovery should take place in the case when a Facilitator-Forwarder terminates.

To examine this case, I started the data source, after which I started 3 hosts on a separate LAN. The first of these hosts designated itself to be both the Facilitator and Forwarder of data,

[12] Bandwidth measurement is done with the use of the same method that ESM uses to measure bandwidth levels; a print statement is provided every 2 seconds which gives the average bandwidth level in kbps that has been maintained over the previous 2 seconds.

while the second and third hosts tuned into the LAN Multicast address to receive channel data. After waiting 5 seconds to allow the hosts to stabilize, I terminated the Facilitator/Forwarder, after which the second host would replace it as its backup. During the 40 second period encompassing this termination, I examined the bandwidth levels of the third host that was receiving data from LAN Multicast.

C. Facilitator Election

To test bandwidth levels during the election process, I started a broadcast on a separate network, similar to the previous case. I then started four hosts on this network and waited 5 seconds to allow the hosts to stability. Afterward, I terminated the designated backup immediately followed by the Facilitator/Forwarder. The result is that the remaining two hosts began the Facilitator election process. Eventually, the third host was designated as the new Facilitator, because it had a lower ID. During this election process, I examined the bandwidth levels of the fourth host that was receiving data from LAN Multicast.

6.2 Results

The results from bandwidth analyses verify that this system is not only robust and resilient, but it is also efficient and economic in its consumption of bandwidth. Let us examine the results in each of the previously described cases.

6.2.1 Normal Case Bandwidth

During the 40 second interval in which the host tuned into the LAN Multicast address, data flow was solid. Within two second, the initial flow of data rose to 399kpbs, remaining at an average of 399.6kpbs for the duration of the test (Chart 1).

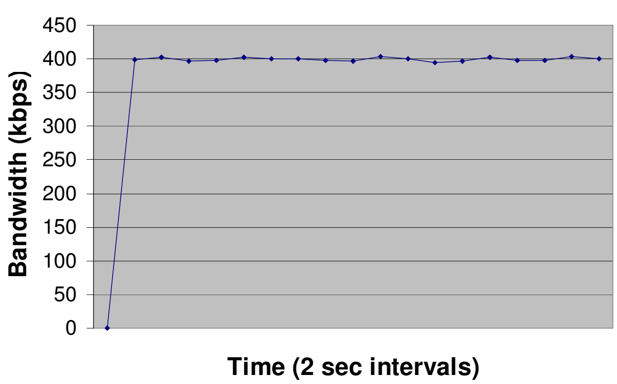

Chart 1. Normal case bandwidth from initialization on

From this information, we can verify that the system achieves a successful rate of data transfer on the LAN, as the average data receipt rate is 99.9% of the transmission rate. We are thusly now concern with cases in which this transmission rate may be interrupted.

6.2.2 Risk Analysis

As stated previously, we are concerned with Facilitator and Forwarder termination cases, are these are the contexts in which data flow has the potential of being interrupted or be otherwise adversely affected.

A. Abnormal LAN Facilitator Termination

As was hypothesized based on the fact that the LAN Facilitator is not responsible for data propagation, the data rate of hosts tuned into the LAN Multicast address was not affected by abnormal LAN Facilitator termination.

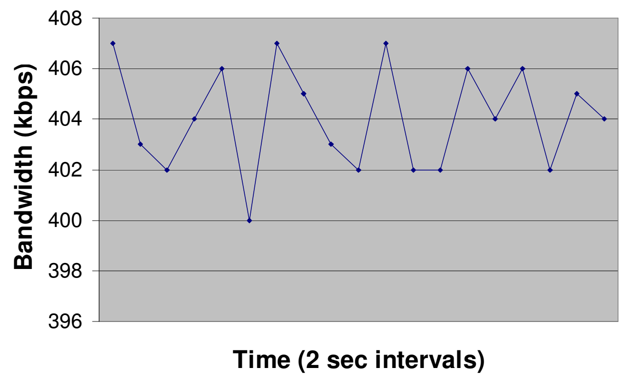

Chart 2. Abnormal Facilitator termination bandwidth

The data receipt levels recorded remained between 400 and 407kbps during the entire duration, never falling below the 400kpbs line (Chart 2). As such, no data loss occurred during the termination and subsequent backup replacement.

B. Abnormal Forwarder and Facilitator Termination

During this transition, from about, the bandwidth levels changes slightly, but it is negligible when put in the context of the bandwidth transmission rate. The bandwidth level fluctuated between 392kbps and 400kbps, averaging 394.9kbps, or 98.7% of the rate of transmission (Chart 3).

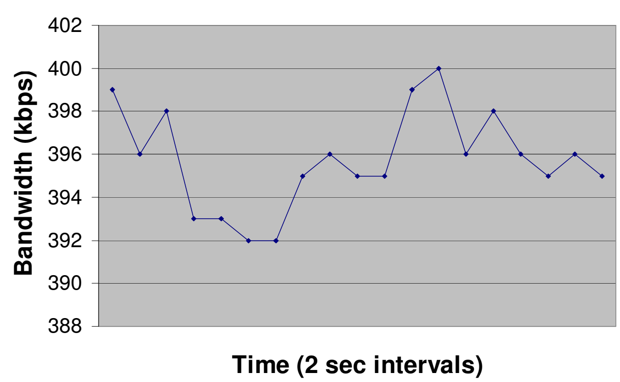

Chart 3. Facilitator/Forwarder termination bandwidth

The lowest point in data transmission overall during this time was at 392kpbs, or 98% of the rate of transmission. The excellent performance of this case demonstrates clearly that the system is resilient to abnormal Forwarder and Facilitator collapse.

C. Facilitator Election

The likelihood of this case happening is low, because it requires both the Facilitator and its Backup to terminate within a second of each other. Regardless, data verifies that this case is detrimental to bandwidth levels during the time in which it occurs (Chart 4).

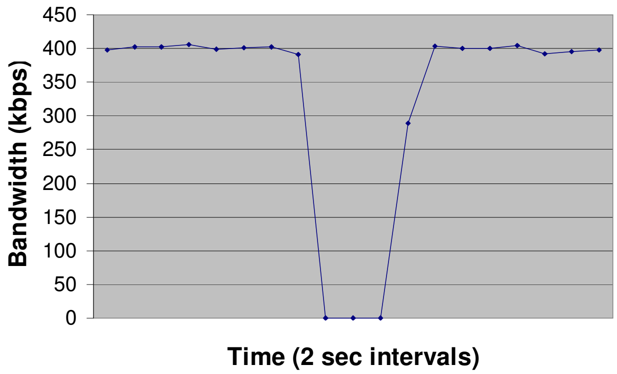

Chart 4. Facilitator election process bandwidth

The data flow before and after the Facilitator election process remained steady near or at 400kbps. However, during the 6 seconds for which there was neither a LAN Facilitator or LAN Forwarder designated, no data was being received. After this 6 second interval when the new Facilitator was elected and established a Forwarder, data rates returned to the range 400kpbs again.

Fortunately, there is a data buffering mechanism in the ESM video/audio playback with allows for bandwidth drops like this. It is not guaranteed, however, that all of these bumps will not be noticed. Given that the duration of this drop is 6 seconds, if less than 6 seconds of video has been buffered but the client, the user will notice a pause in video during this time.

7. SUMMARY AND FUTURE WORK

In this paper, I have reported on the design, development, implementation, and analysis of a peer-to-peer system based on the benefits of LAN Multicast and End System Multicast. To my knowledge, ALHM is the first of its kind to integrate these two protocols into a system that has the ability to support multiple broadcast channels over the globe with an O(1) cost to data sources and an O(1) level of broadcast traffic for each channel on participating LANs.

As I mentioned earlier, because LAN Multicast comes into effect when 2 or more hosts are using ESM on the LAN, the threshold number of users upon which a LAN transitions from ESM to LAN Multicast is 2. It may be beneficial in certain situations to have a higher threshold. Consider the following scenario: A network contains 1000 users broadcasting 1000 channels, and for each channel, only 2 users are tuned in. Because LAN Multicast is implemented on many systems by broadcasting Multicast packets to all hosts (at which point hosts that are not tuned in ignore these packets), the result is that the use of LAN Multicast in this case is overkill.

It is very difficult to determine, however, when LAN Multicast is overkill and when it is indeed the best route to take; variables such as LAN size, LAN topology, host capabilities, and channel data rates all play an important factor in determining whether standalone ESM would be more or less optimal than LAN Multicast in these cases when large numbers of channels with few viewers occurs. Since the system is still working on obtaining many data publishers to do broadcasts, this issue has quite a bit of time before it will need to be addressed.

Nevertheless, Adaptive LAN-to-Host Multicast goes above and beyond the needs of real content publishers and spectators as a cost-effective high-quality approach to Internet broadcasting. At one point in time, the only organizations capable of globally pervasive broadcasting mechanisms were TV and Radio broadcasting corporations. This is no longer the case, as users are now empowered to create their own global broadcasting channels and use p2p technology to achieve their goals for free. The world that may result from technologies such as these is a world in which every person has the power to shape the global media community. In this new world, the most-viewed content will no longer be as such based on the power of the organizations that do the broadcasting, but rather it will be based on the quality of the content itself.

8. ACKNOWLEDGMENTS

I would like to give my most sincere thanks to the entire Carnegie Mellon End System Multicast Group for their assistance in integrating and successfully deploying this system with CMU ESM. Were it not for you all, ALHM would not be a reality.

9. REFERENCES

[1] Y. Chu, A. Ganjam, T.S.E. Ng, S.G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang. *Early Experience with an Internet Broadcast System Based on Overlay Multicast*. Proceedings of the USENIX Annual Technical Conference, June 2004.

[2] B. Zhang, S. Jamin, and L. Zhang. *Host Multicast: A Framework for Delivering Multicast to End Users.* Proceedings of IEEE INFOCOM, June 2002.

[3] Cisco, Inc. *Internet Protocol Multicast.* 2006. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ipmulti.htm#xtocid1

[4] Y. Chu, S.G. Rao, and H. Zhang. *A Case for End System Multicast*. Proceedings of ACM SIGMETRICS, June 2000.

[5] BitTorrent. 2006. http://www.bittorrent.com/.

[6] Gnutella. 2006. http://www.gnutella.com/.

[7] W. Wang, D. Heiler, S. Jamin, and L. Zhang. *Overlay Optimizations for End-host Multicast*. Proceedings of Fourth International Workshop on Networked Group Communication (NGC), October 2002.

[8] End System Multicast toolkit and portal. 2006. http://esm.cs.cmu.edu/.

[9] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang. *Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture*. Proceedings of ACM SIGCOMM, August 2001.

[10] S. Floyd, M. Handley, J. Padhye, and J. Widmer. *Multicast Routing in Internetworks and Extended LANs*. In Proceedings of the ACM SIGCOMM, August 2000.

[11] J. Liebeherr and M. Nahas. *Application-layer Multicast with Delaunay Triangulations*. IEEE Globecom, November 2001.