

# A Simulation Model for Hybrid Multicast

Dhaifallah Alwadani, Mario Kolberg  
Computing Science and Mathematics  
University of Stirling  
Stirling, Scotland  
{dba,mko}@cs.stir.ac.uk

John Buford  
Avaya Labs Research  
Basking Ridge, New Jersey, USA

## Abstract

*In order to achieve one-to-many data delivery on the internet, native multicast is used and implemented in some parts of the Internet. On the other hand, application layer multicast (ALM), which uses P2P overlays (overlay multicast, OM), can be used to create multicast trees and deliver the data at the application layer. Despite Native Multicast being more efficient than Application Layer Multicast, it is not deployed widely.*

*A hybrid multicast protocol has been proposed that opportunistically combines overlay multicast protocols and native multicast protocols to create and maintain hybrid multicast trees. The design for hybrid multicast trees leverages the AMT multicast tunneling protocol.*

*It is expected that this hybrid approach will improve both efficiency and availability of multicast. This paper presents a simulation model for the Oversim/Omnet++ simulation framework to evaluate the performance of the hybrid multicast approach. Our model combines both a scalable overlay and a detailed network layer model that includes routers with native multicast support.*

## I. INTRODUCTION

Different Internet applications, most notably IPTV and conference calls, rely on distributing content in a one-to-many or many-to-many fashion. Multicast is widely seen as important for such applications to reduce their bandwidth requirements. According to Cisco's Visual Networking Index [8] 91% of Internet traffic is expected to be video.

Multicast can be offered both at the network layer and also application layer. The former depends on routers forwarding and replicating multicast messages. This is termed Native Multicast. A major technique in this domain, Host Group Multicast, employs routers to form a spanning tree for each multicast group. Routers maintain state of the groups. Alternatively, for Multidestination Routing [15], group information is included in the multicast messages and routers make routing decisions on all addresses in the messages.

On the other hand, Application Layer Multicast (ALM), uses hosts rather than routers for forming the spanning tree and replicating and forwarding the multicast messages. Thus ALM does not require multicast support in the

network routers. On the downside, it is not as efficient as Native Multicast. The approach presented in this paper uses Overlay Multicast (OM), a type of Application Layer Multicast which uses a peer-to-peer overlay network to form multicast trees.

Currently, the multicast support by the routers as required by Native Multicast is not universally available across the Internet. Indeed multiple incompatible native multicast approaches co-exist. This has led to the formation of so-called multicast islands. Thus even if the spread of multicast-capable routers is increasing, due to different approaches deployed, the problem of multicast islands will continue to exist. Thus a solution connecting the various Multicast Islands and bridging between different NM techniques is needed. Application Layer Multicast together with tunneling approaches can be used for this purpose, leading to Hybrid Multicast solutions. By doing so, the benefits of both techniques, i.e. the greater efficiency of Native Multicast and the readily availability of ALM can be gained. While Hybrid Multicast has significant benefits, it has not been given sufficient attention by the research community [11]. This paper aims to address this shortcoming by analyzing and evaluating a framework for Hybrid Multicasting using peer-to-peer overlay by integrating Native Multicast (NM), Automatic Multicast Tunneling (AMT) and Application Layer Multicast (ALM).

A basic design for a hybrid protocol has been defined by Buford in [2][3]. It is based on integrating AMT tunneling mechanism for NM[17] with OM protocol mechanisms. Here we propose an approach to simulating and evaluating the performance of this hybrid multicast protocol.

Evaluating hybrid multicast protocols is difficult for two reasons. The simulation environment must combine both a scalable overlay and a detailed network layer that includes routers with NM support. Suitable metrics are needed to compare the tree quality with pure OM trees. This paper presents our approach to simulating and evaluating hybrid multicast protocols. The authors have previously studied a related problem with parallelizing overlay messaging with multi-destination multicast routing [4].

This paper presents our approach to simulating and evaluating hybrid multicast protocols. We present a simulation framework which includes a model of the underlay network integrated with a model of the overlay.

For the underlay, we present an implementation of an AMT module which we use to tunnel multicast traffic between networks with IGMPv2 support. This is the first simulation work to incorporate the AMT architecture into a hybrid multicast framework. Our overlay model includes peer-to-peer and overlay multicast support. For the experimentation in this paper we use the Scribe algorithm. Thus our framework is different to most overlay simulations in that it combines an overlay with a packet level model supporting multicast and tunneling. We present simulation results comparing message load for an AMT backbone with an ALM backbone in hybrid multicast demonstrating the suitability of our hybrid multicast approach. The authors have previously studied a related problem with adding ALM support to RELOAD [18].

## II. RELATED WORK

Tunneling multicast over unicast-only network is an impotent aspect of the solution. These tunnels play a vital role in the performance of multicasting. These techniques can be classified by the way these tunnels are built:

- Explicit Tunneling: In this case, the tunnels need to be designed, created and maintained manually.
- Without Explicit Tunneling: Tunnels are created and maintained automatically.

### A. Multicast with Explicit Tunneling

With Explicit Tunneling, isolated multicast islands are interconnected using manual tunnels such as Generic Routing Encapsulation (GRE). This is a manual approach carried out by network administrators and hence does not scale. Furthermore, in the case of Multicast over GRE, each multicast packet is encapsulated in a GRE header[9]. Thus the layer 4 information is hidden to the routers connecting the end-points of the tunnel. However, layer 4 header information is needed for doing QoS and trace shaping.

### B. Multicast without Explicit Tunneling

With this approach, tunnels will be created and maintained automatically. The shape and topology of these tunnels will change dynamically as the underlying network changes. This could be done using different techniques such as peer-to-peer overlays or AMT tunnels.

Automatic Multicast Tunneling (AMT) [17] provides a way to build the unicast tunnels between multicast islands when needed without explicit configuration. AMT defines two main components: AMT Gateways and AMT Relays. Tunnels are established between pairs of a gateway and a relay using UDP (User Datagram Protocol) encapsulation. The multicast tunnels will keep the layer 4 header in the packet allowing for better QoS. One drawback of using AMT is the lack of some important features such as resource locating and management. [10] attempts to solve this issue by extending RELOAD to work with AMT.

In [11], the authors show a way to combine and connect Native Multicast tunnels using overlay data distribution. The paper also proposes two different protocols to solve the problem of Island Multicast:

**Centralized Island Multicast CIM** This protocol is suitable for small sized groups that have many-to-many communication and high bandwidth requirement e.g. multi-party conference calls. Here, there is a central server to build and maintain the spanning tree.

**Distributed Island Multicast DIM** This protocol is suitable for large groups where scalability is required. In DIM, hosts in the same multicast island elect a unique leader. The leader node will be responsible for constructing the delivery overlay.

As has been proposed in [12], Universal Multicast (UM) provides a way to connecting multicast islands using dynamically build unicast tunnels. UM allows for multiple connections for an island. Doing so will allow for an improvement in speed for large islands. Inside each multicast island, one or more Dedicated Members (DM) are elected to natively deliver the multicast to the island using Native Multicast. Also, the authors of Universal Multicast have proposed a protocol for intra-island multiple-DM management protocol [13]. This protocol is called Host Group Management Protocol HGMP. HGMP is concerned with electing peers to be DM in an island dynamically and using multiple DMs together for load-balancing purposes.

A protocol for connecting multicast islands has been proposed in [14]. This does not depend on the native multicast protocol. However, there are issues of scalability and its incompleteness compared to other solutions.

## III. HYBRID MULTICAST

### C. Framework

Figure 1 shows an example native topology divided into five regions. Peers in regions with NM support are labeled  $P_N$ . Peers in regions without NM support are labeled  $P_A$ . A peer which supports the AMT protocol is labeled  $P_{AMT-GW}$ . The multicast tree uses P2P overlay connections to any  $P_A$ . Within a NM region, peers in the same hybrid tree use NM. Between NM regions, peers connect via the  $P_{AMT-GW}$ , using the AMT protocol which encapsulates NM packets.

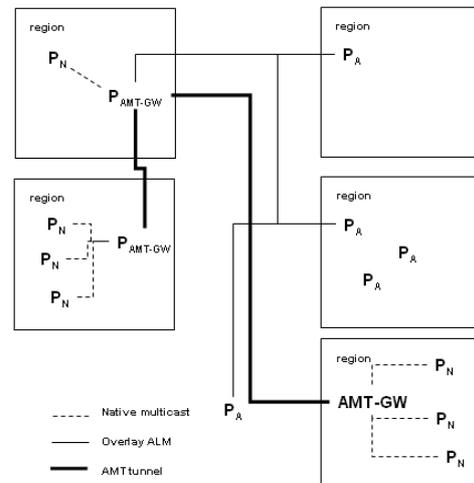


Figure 1 Example of network divided into 5 regions.

Consequently, the proposed model will join application layer trees with native multicast trees. At the application layer the approach makes use of P2P overlays opportunistically combining ALM trees and native multicast where available and linking multicast islands using AMT.

The proposed model takes advantage of cross layer awareness combining information from different layers to make decisions. Decisions are needed for Islands Discovery and Nodes Election Mechanisms. By accessing the information available in lower layers (Network and Data Link), ALM peers are aware of the status of the underlay including the support of native multicast and the existence of other peers in the same multicast domain. Peers joining the same ALM group who are also located in the same native multicast island communicate more efficiently using native multicast for intra-island communication.

In the approach, the peers optimize the way that they are connected to the multicast tree. Hosts inside an island will elect which nodes will directly connect to the ALM tree. This elected peer will be responsible for relaying the information between the ALM and Native Multicast trees. Thus, the approach will distinguish the following three types of communication: Intra-Island, Inter-Island and Overlay Communications. The proposed hybrid approach improves both, efficiency and availability of multicast communication as well as its Scalability and Adaptability.

#### IV. SIMULATION MODEL

##### D. Simulator Selection

There are many simulators but we have chosen OMNeT++ with the INET model together with Oversim for a number of reasons including:

**Extensible and Modular:** This is a must have feature since there currently is no implementation for AMT in any existing simulator. Omnet++ is open source and hence freely extendible. One of the most important features of OMNeT++ is its modular design. OMNeT++ supports simple modules and compound modules. The latter groups other modules together and describes how these sub-modules are connected. Compound modules are written using the NED Language which is topology description language. On the other hand, simple models are written in C++ and implement protocol logic. Omnet's hierarchal structure helps reusing some of the existing code in new protocol models and makes it possible to share modules in the open source community.

**INET model:** This model provides a wide range of models of several Internet protocols which include IGMP.

**Oversim** [6]: is built on top of Omnet++ and provides a framework to simulate peer-to-peer overlays protocols. It includes models of many common protocols. It can achieve larger network sizes with 10s of thousands of nodes and is able to interface with INET to include specific underlay behavior in the simulation.

##### E. Changes to the Simulation Environment

Currently, there is no working AMT components and support in OMNeT++, hence it needed to be added to enable hybrid multicast simulations. The components implemented are in accordance with the AMT IRTF draft [17]. The following components have been implemented:

**AMT Gateway** This component implements an AMT Gateway.

**AMT Relay** This component implements an AMT Relay.

**AMT Network Messages** This implements different kinds of messages needed in AMT operation.

**AMT Gateway Host** This host will be able to connect to an AMT Relay directly.

**Simulation Signals** We will need to add some signals to collect relevant results.

We discuss details of these components below.

##### F. AMT Gateway

The AMT Gateway consists of a number of components. These are shown in Figure 2.

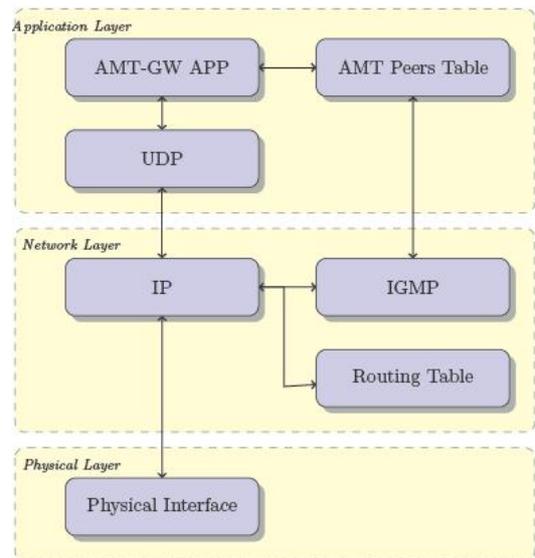


Figure 2: The AMT Gateway Module.

Most of the logic for the AMT Gateway is contained in the AMT-GW App. This App uses UDP to communicate with other node in the network. The AMT Peers Table stores addresses of known peers and hosts.

```

Require: msg.
1 if msg = AMT_Relay_Advertisement then
2:   addRelayToAMTPeerTable msg.source
3: else if msg = AMT_Query and msg.group ∈ AMT_Peer_Table then
4:   replay AMT_Membership_Update
5: else if msg = AMT_Data then
6:   for all peer ∈ AM_table do
7:     forward msg:encapsulatedData
8:   end for
9: else if msg = IGMP_Membership_Update then
10:  AddHostToAMTPeerTable msg.source
11: end if

```

Algorithm 1: AMT Gateway Component Algorithm

### G. AMT Relay

The AMT Relay is the counterpart of a Gateway and has a very similar structure to the gateway model. An overview of the Relay component is shown in Figure 3.

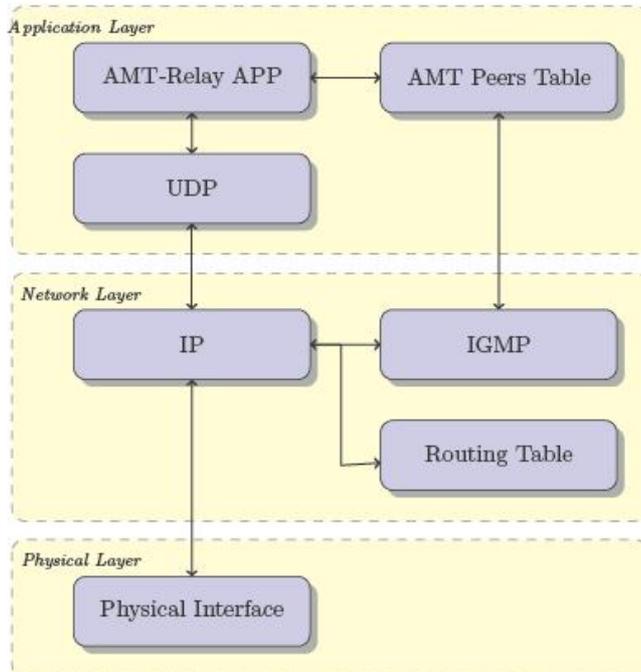


Figure 3: The components of the AMT Relay module.

Like the AMT Gateway component, the AMT Relay behavior is controlled by the AMT-Relay App that uses UDP for communication. The App listens and waits for a connection request from an AMT Gateway. Also it listens for multicast messages and forwards them as appropriate.

```

Require: msg .
1: if msg = AMT_Relay_Discovery then
2:   replay AMT_Relay_Advertisement
3: else if msg = AMT_Request then
4:   replay AMT_Membership_Query
5: else if msg = AMT_Membership_Update then
6:   add_to_AMT_table msg:igmp:group
7: else if msg ∈ Multicast_Message then
8:   for all peer ∈ AMT_table do
9:     forward AMT_DATA:encapsulate(msg)
10:   end for
11: end if
  
```

Algorithm 2: AMT Relay Component Algorithm

Similarly to the design of the Peer table in AMT Gateways, the Peer table in an AMT Relay keeps track of AMT peers in the network. However, in this table, we only keep track of AMT Gateways and the multicast groups that they are interested in. As the AMT specification draft does not offer a dynamic way to configure them they need to be filled in manually. However, in hybrid multicast the overlay can be used to interconnect the relays and gateways. We implemented the following process:

1. Each AMT device joins the ALM Overlay (we used Scribe based on Pastry).
2. Each AMT-GW will hash the multicast group address to generate an overlay key. Then, the AMT-GW will subscribe to this key.
3. When an AMT-Relay receives native multicast traffic, it will hash the group address to get the corresponding overlay key. It then sends the data to that ALM group.

### V. EXPERIMENTATION

In order to test the proposed hybrid multicast framework and our implementation on Omnet++/Oversim, we have setup a number of experiments. Our experiments use a 1000 node network with a multicast group size ranging from 200 up to 1000 nodes. A source node sends multicast traffic at 1 packet every 5 seconds. In the experiments we use Scribe as the ALM protocol together with our AMT implementation and the IGMPv2 implementation of INET. We use a setup which has 50 networks which can be used as multicast islands which are all interconnected by a backbone network of routers. We have setup the following configurations:

**ALM only:** Scribe is used to manage the group and distribute the multicast data. There is no native multicast support anywhere in the network.

**Native Multicast only:** IGMPv2 is used to manage the groups and distribute multicast data. The 50 networks and the backbone routers support IGMP.

**AMT + native multicast:** The 50 networks support IGMP, but the backbone is unicast only. This essentially models native multicast islands. AMT devices, which are located in every island, will tunnel the traffic to other AMT devices at other islands across the backbone. The AMT devices there will multicast the traffic within the island.

**Hybrid overlay:** Again, the 50 networks support IGMP, but the backbone is unicast only. In this configuration, the AMT devices use Scribe to discover each other and forward multicast traffic between islands.

As can be seen in Figure 4, when simulating the ALM only configuration, the results show an increase of the message load in both the islands and also the backbone. In the figure the x-axis shows the network size and the y-axis the number of packets in the network. This is exactly as expected. While straightforward to deploy, the efficiency of ALM is limited.

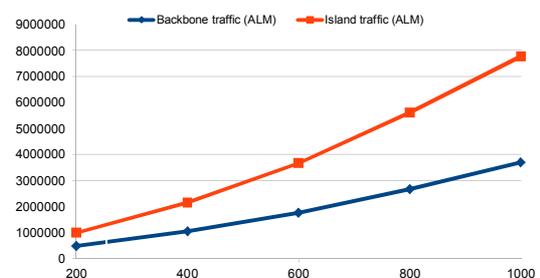


Figure 4: Results of using ALM only.

Figure 5 shows the performance when using native multicast in the islands and also on the backbone. This yields a much better performance. As the network size grows, the load on the backbone remains static. The load within the islands is considerably lower than with ALM. It still grows as the messages need to be distributed to all the nodes, but at a much lower level than with ALM.

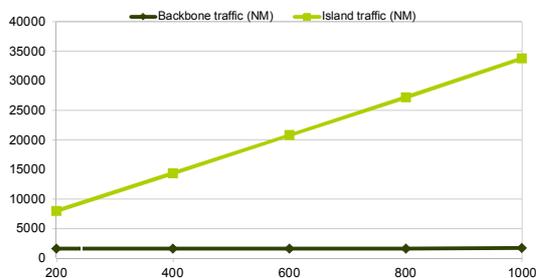


Figure 5: Results for a native multicast only.

Figures 6 and 7 show the network load when the native multicast enabled islands are linked with a unicast backbone. Figure 6 shows the load when the islands are connected using AMT, whereas Figure 7 shows the results when ALM is used for this purpose. For both configurations, the island load is almost identical to the native multicast load shown in Figure 5. This is to be expected as the load within an island does not change.

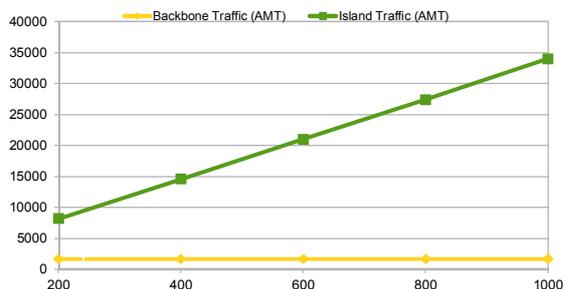


Figure 6: Linking native multicast islands with AMT.

The backbone load is slightly lower when AMT is used for linking the islands. Clearly AMT is more efficient than ALM. However, AMT requires network support whereas ALM is easier to deploy.

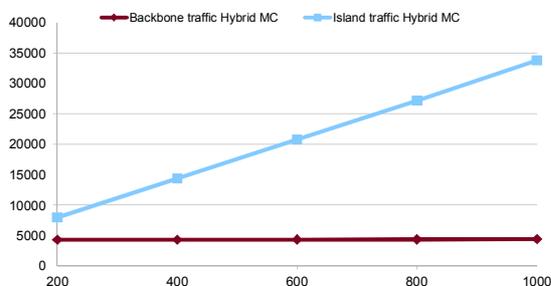


Figure 7: Hybrid multicasting.

Combining ALM and AMT for linking native multicast enabled islands has a number of advantages. For one, AMT

can use the ALM infrastructure of pair gateways and relays. This is especially of importance in scenarios where the number of AMT gateways is large and the location of AMT relays is not known. Secondly there may be AMT enabled islands which are not linked to the ALM. Conversely, there may be islands which have a link to the ALM, but are not AMT enabled. The joint approach allows for nodes in both types of islands to be included in the overall multicast tree.

## VI. CONCLUSION

We have presented a simulation framework for OMNeT++ and Oversim to simulate hybrid multicast. In doing so, we have implemented AMT relay and gateway components in OMNeT++ and linked the IGMP enabled multicast islands using both AMT and ALM implemented in Oversim. In this model, native multicast is used opportunistically wherever possible. That means using native multicast within multicast islands, AMT for connecting independent islands and ALM for connecting nodes to the multicast tree which do not have any native multicast connectivity. We have presented results, comparing the message load on the backbone for AMT with ALM which demonstrate the validity of hybrid multicast.

## VII. REFERENCES

- [1] J. Buford, S. Kadadi. SAM Problem Statement. Dec' 06. Internet Draft draft-irtf-sam-problem-statement-01.txt, work in progress.
- [2] J. Buford. Hybrid Overlay Multicast Framework. IRTF SAM RG. draft-irtf-sam-hybrid-overlay-framework-02. March '08, Work in Progress.
- [3] J. Buford. SAM Overlay Protocol. IRTF SAM RG. draft-irtf-sam-overlay-protocol-00.txt, Feb 2008. Work in progress.
- [4] J. Buford, A. Brown, M. Kolberg. Exploiting Parallelism in the Design of Peer-to-Peer Overlays. J. Computer Communications. Special Issue on Foundations of Peer-to-Peer Computing Vol 31/3, Feb. 2008, pp 452-463.
- [5] I. Baumgart, B. Heep, S. Krause. OverSim: A Flexible Overlay Network Simulation Framework, IEEE Global Internet Symposium 2007.
- [6] A. Varga. Omnet++ community site. <http://www.omnetpp.org>
- [7] Y.-H. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. In Proceedings of ACM SIGMETRICS, June 2000.
- [8] CISCO. Cisco Visual Networking Index : Forecast and Methodology, 2012 – 2017, White Paper, pages 1-10, 2013.
- [9] Yu Gu, Lee Breslau, N. Duffield, and S. Sen. GRE Encapsulated Multicast Probing: A Scalable Technique for Measuring One-Way Loss. In 2008 IEEE INFOCOM, pages 1651-1659. IEEE, 2008.
- [10] M. Kolberg and J. Buford. Application Layer Multicast extensions to RELOAD. 2011 IEEE Consumer Communications and Networking Conference (CCNC), pages 1083-1087, January 2011.
- [11] Xing Jin, KL Cheng, and SHG Chan. Island multicast: combining IP multicast with overlay data distribution. IEEE Transactions Multimedia, 11(5):1024- 1036, 2009.
- [12] Sugih Jamin, Lixia Zhang, and Beichuan Zhang. Host multicast: a framework for delivering multicast to end users. 21<sup>st</sup> Joint Conference of the IEEE Computer and Communications Societies, 3:1366-1375, 2002.
- [13] Beichuan Zhang, Sugih Jamin, and L Zhang. Universal IP multicast delivery. Int'l Workshop on Networked Group Communication, 2002.
- [14] Juyoung Park, Dae Young Kim, Shin Gak Kang, and Seok Joo Koh. Multicast delivery based on unicast and subnet multicast. IEEE Communications Letters, 5(4):181-183, April 2001.
- [15] L. Aguilar, Datagram, Routing for Internet Multicasting, ACM Sigcomm 84, March 1984.
- [16] Andreas Varga and Rudolf Hornig. An overview of the OMNeT++ simulation environment. 1st Intl. Conf. on Simulation, 2008.
- [17] G. Bumgardner. Automatic Multicast Tunneling (AMT). Internet Draft draft-ietf-mboned-auto-multicast-17, Work in progress. Apr. 2014.
- [18] M. Kolberg (Ed.), J. Buford. Application-Layer Multicast Extensions to REsource LOcation And Discovery (RELOAD), RFC 7019, IRTF, 2013.