

Comparison and Evaluation of Application Level Multicast for Mobile Networks

Ingo Juchem

E-mail: ijuchem@cs.uni-goettingen.de

Abstract

This essay addresses two different aspects of multicasting in mobile networks, namely application-level multicast and its implementation using peer-to-peer overlays. Application-level multicast, as against IP multicast, although being a newer approach, will prove to be inferior in terms of performance in mobile networks. Implementing it using peer-to-peer overlays will prove to be not beneficial.

1. Introduction

Multicast has been rediscovered as a means to address a group of nodes in a mobile network. It is believed that it poses less link stress to a network than broadcast or unicast. Unfortunately, *IP multicast (IPM)*, being introduced at a time where mobility did not exist in networks, was not designed for today's requirements in mobile networks – be it cellular, wireless or sensor networks.

Application-Level multicast (ALM) is a new idea where the routing is moved from the network layer (where IPM is working on) to the end hosts or *nodes* in a mobile network themselves.

In addition to this, several approaches have been deployed to implement ALM and to use it in non-wired networks. One main approach being structured *Peer-to-peer overlays* which claim to be scalable in and support large networks with many different end hosts.

This essay presents different aspects in this field. It is organized as follows: Section 2 presents the motivation for applying multicasting and the problems in today's mobile networks, followed by an introduction of IPM as well as ALM in general. After that will be a description of peer-to-peer overlays. Chapter 3 will contain the evaluation of two simulations which will deal with a comparison of IPM and ALM and the benefit of implementations of peer-to-peer overlays used for building ALM. We will see that ALM is not only no solution for problems in mobile networks but even that IP multicast outperforms ALM in all aspects. Even with peer-to-peer overlays being used performance regarding link stress, delay and robustness is still worse with ALM than with IPM. After chapter 4, which will describe the conclusions from the evaluations, I will go beyond the previous work and try to identify gaps and limitations in the two works and identify the strengths and weakness of them.

2. Motivation and Background

In mobile networks, as opposed to stationary networks, nodes usually do not have a fixed position. Especially in wireless LAN and cellular networks participating nodes are fast moving parts of the network. This high mobility poses a new challenge for the structure and the protocols (e.g. Mobile IPv6) working together to make communications between the nodes possible, reliable and secure. Today's networking protocols have not been designed to handle fast moving nodes in a mobile environment, as at the time they were introduced today's possibilities in networking were out of scope.

In general, with high mobility comes high network usage as the moving nodes and routers have an increased need for communication. For packets to traverse the correct paths and to be able to reach their destination, routers need to keep constantly updated tables for finding the optimal path between nodes. Now, if participating nodes move around in the network at high speed (imagine using a cellphone in a high-speed train) these tables need to be updated more frequently at higher speed than in classic, fixed, network environments. Also, each node in the mobile network needs to address its location to the corresponding router(s).

To be able to deal with this challenge, the idea of addressing a group of nodes instead of every single node in a network was brought to light - *multicasting*. The first idea was using *IP multicast*. We will see that this traditional approach is not capable of solving the problems in mobile networks. Instead, a new approach called *Application Layer Multicast (ALM)* was introduced.

2.1 IP Multicast

In *IP multicast*, which is working on the network layer, nodes use a special way to address many other nodes in their network. This is called one-to-group addressing. It is a protocol for transmitting packets from one single source to many destinations. Routing of packets in IP multicast is handled by the network routers on the same layer. Other ways to address a destination are unicast and broadcast. Unicast, where packets are transmitted from one source to one destination, is a one-to-one addressing. Broadcast is a way to address all other nodes in a network, a one-to-all addressing. Unfortunately, IP multicast is a too complex to be used and incapable of working in mobile networks. Therefore, a new approach on a different layer was introduced. General information about IP multicast for mobile network can be found in [1].

2.2 Application Layer Multicast

Application Layer Multicast (ALM) is a new idea for multicasting. It is working on a layer above network layer, the application layer. ALM was designed for easier use than IP multicast but still not for mobile networks.

The idea of ALM is to shift the management of groups and packets as well as the routing from IP routers on the network layer to the end hosts on the

application layer. Packets on the network layer are transmitted using standard unicast. By this shifting, the physical topology of the network is completely hidden from the routing algorithms. This is done by constructing an overlay on the current network. We will see later how this is done exactly.

By constructing this overlay and hiding the physical topology of the underlying network from the nodes, ALM claims to be independent from the characteristics of the underlying network and therefore disregard node movement.

However, mobility in a network has a negative impact on the ALM approach. This is caused by ALM's overall design which faces difficulties in non-wired networks for which it was not designed. Instead, it is mainly concerned with failure of a single or multiple nodes in the network, not with their mobility. A mobile network can consist of many different nodes, each with their own characteristics and is therefore, in contrast to wired networks, heterogeneous. These nodes need to be cared for singly, with each having its own capabilities like battery power, which ALM can not deal with.

The main problem for ALM in mobile networks is that it is designed to heavily depend on the end hosts (remember that routing is shifted to the end hosts on the application layer), which will be far less robust than in a wired network.

We will see that ALM is no solution for the above mentioned problems in mobile networks, and does not perform better than IPM. But first we will have a look at a way to implement ALM, the *structured Peer-to-peer overlays*.

2.3 Structured Peer-to-peer overlays

In general, peer-to-peer (p2p) overlays can be used to implement internet-scale application level multicast. They provide an efficient way for packet routing by assigning parts of the namespace to nodes and supply nodes with the needed functionality to communicate with each other. The following example shows how a namespace can be divided into subgroups:

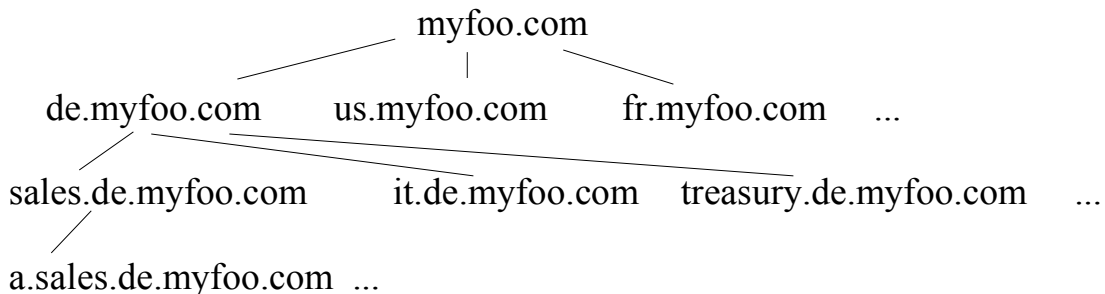


Fig. 1: “An example of dividing a namespace in groups”

In the above example, the domain or namespace “myfoo.com” is divided into groups. Let us assume myfoo.com is our company. We have dependencies in many countries of the world, therefore we divided it into subdomains for each country: de.myfoo.com for our German dependency, fr.myfoo.com for our French and so on. Each of our dependencies has itself several subdivisions (groups) like

it, sales, treasury etc. which again can be subdivided into many lower level groups in our hierarchy. We can easily see that this is structured with several levels in the hierarchy.

There are several approaches of p2p overlays, for example CAN [2], Chord [3], Pastry [4] and Tapestry [5], each using a different approach for multicasting and routing. Routing algorithms for p2p can be divided into two classes, divide-and-conquer (used by Chord, Pastry and Tapestry) and routing in a Cartesian hyper-space (used by CAN). All these algorithms are scalable and self-organising.

There are two basic methods for implementing multicast in these approaches, namely tree-building and flooding. The tree-building method uses a single overlay on the network and builds a tree for each of the groups in the structured hierarchy. This tree is used to transport the multicast messages to nodes. The flooding method creates separate overlays for each group and uses the routing information known to each group to broadcast messages to each node in the overlay.

However, current implementations of peer-to-peer overlays show several problems. All of them are highly complex with a large set of adjustable parameters (for example Landmark-based placement, network aware routing), each being beneficial as well as having drastic positive and negative influence on performance. Each implementation uses a different approach so it is unsure if and which approach will work in a specific environment. This is especially caused by the lack of evaluations on the performance in mobile networks of the 4 possible combinations of multicast and routing algorithms (tree-building, flooding, divided-and-conquer, Cartesian hyperspace).

Therefore, the following paragraph will describe both approaches for implementing multicast in CAN and Pastry, namely the flooding and tree-building algorithms. These approaches will later be evaluated in this essay.

2.4 Approaches for peer-to-peer overlay networks

For the later evaluation, it is necessary to have a look at the two different approaches for implementation of p2p overlays regarding multicast as well as routing algorithms.

The *Content addressable network (CAN)* organizes – for routing purpose – the nodes in a d-dimensional hypercube. There, each node takes the responsibility for a small portion of the network and keeps a routing table to its nearest neighbours. Routing itself is done by forwarding the message to a node closer to the destination in the hyperspace. CAN has several adjustable parameters, which are well described in [6]. Through the evaluation, we will see which of them and their combination can be beneficial for performance.

The *Pastry overlay network* uses a different approach for routing. It assigns a random node ID out of a 128 bit namespace to each node. This node ID's length can be adjusted by a parameter b with a typical value of 4. When N is the number of nodes in the overlay, the expected number of hops from the source to the destination is $\log_2 bN$. Routing itself is done by forwarding the message to a node whose ID is numerically closer to the destination. A special characteristic of

Pastry is that it measures the *round trip time (RTT)* to each node when building the routing tables and therefore exploit network locality when routing.

As described before, there are also two different approaches to implement multicasting in peer-to-peer overlays, *flooding* and *tree-building* (also called *tree-per-group implementation*). In flooding, nodes which desire to join a group in the network need to be provided a lookup function. This requires a distributed name service which can be implemented both using *CAN* and *Pastry*. *CAN* uses a broadcast algorithm for flooding where nodes forward the message to all their known neighbours. *Pastry* also uses a broadcast algorithm. There, a node forwards a message to all other nodes held in its routing table. The tree-based multicast, which will be implemented using *Scribe* [7],[8], a generic application level multicast infrastructure, uses its own approach. Here, a reverse path forwarding is used to build a multicast tree *per group*. Those groups are identified by unique *group IDs*. *Scribe* is a scalable and failure-tolerant decentralized algorithm. More details are available in [8].

To see the performance of these four combinations, it is necessary to evaluate them in a simulation setup. Before doing this, however, we will first see how *ALM* performs over *IPM*. This will be evaluated in the next section.

3. Evaluation

3.1 Application Layer Multicast and IP multicast

To see which approach for multicasting is better in terms of performance, several metrics for measurement have to be introduced first.[9]

There are three important metrics for measuring the overall benefit of IP multicast (IPM) and Application-level multicast (ALM): Network performance, link stress and robustness. For network performance, a metric called *Relative delay penalty (RDP)* was introduced. It is computed as follows:

$$RDP = \frac{ALM \text{ link cost}}{IP \text{ multicast link cost}}$$

Looking at the equation, one can easily see that smaller values for *RDP* mean, that *ALM* is better. Larger values mean that *IPM* is better. In a mobile network, there are 4 components for mobile receivers, two for ALM and two for IPM. In ALM, packets can be transported either by reverse tunnelling (where packets get tunnelled through the home agent) or optimized routing (here the packets go directly to the receiver). For IPM, the receiver can either be in the home network (home subscription) or in a foreign network (remote subscription).

Link stress can easily be measured by the number of identical packets received by the nodes, *robustness* by the data throughput and the number of packets lost.

The model used in this simulation [9] consisted of 500 nodes with 10-20 receivers. First I will describe the results for robustness, then RDP and finally link stress which the simulation showed.

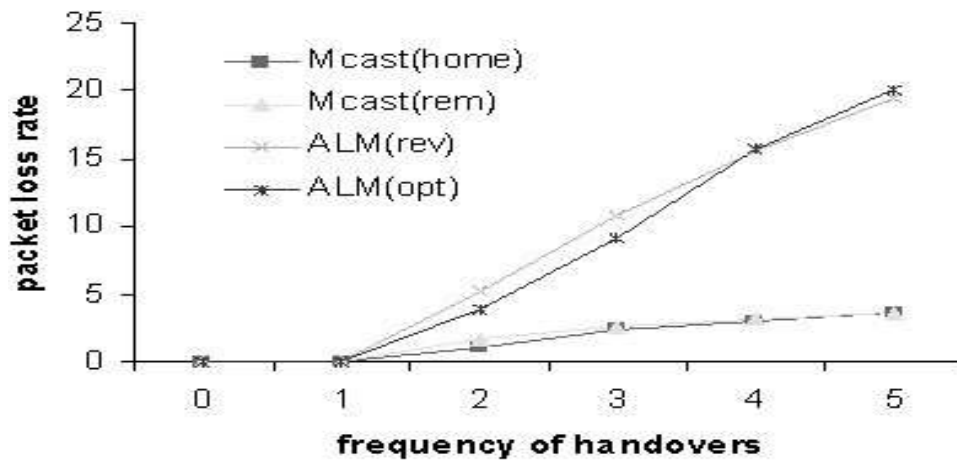


Figure 2: “Packet drop rates for ALM and IPM” [f2]

The x-axis shows the different values for the frequency of handovers whereas the y-axis displays the packet loss rate, which is used for measuring the robustness. For the 4 different components mentioned above we can see that IPM has a low packet loss rate below 4, even with frequent handovers (which is similar to fast movement of the nodes). ALM, however, suffers drastic penalties with increasing movement. We can also see that the drop rate increases much faster with ALM.

Overall, figure 2 shows that the additive path applied by reverse tunnelling in ALM and remote subscription in IPM increases the packet loss rate. ALM performs far worse concerning this metric than IPM.

Figure 3 shows the measurement of the Relative delay penalty, with the group size on the x-axis and RDP on the y-axis:

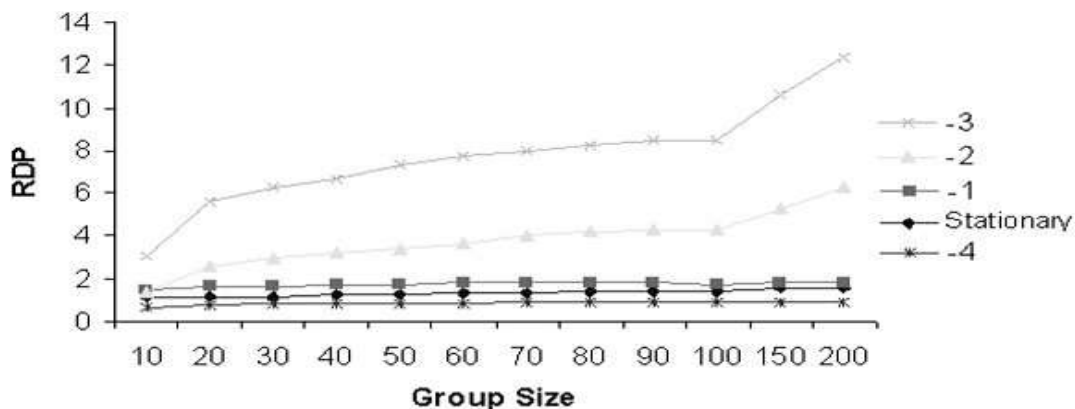


Figure 3: “Effect of mobility on RDP” [f3]

The 5 different curves in the graph are: stationary nodes (stationary), ALM reverse tunnelling over IPM home subscription (1), ALM organized routing over IPM remote subscription (2), ALM reverse tunnelling over IPM organized routing (3) and ALM organized routing over IPM home subscription (4). Here, (1) is similar to fast movement, whereas (2) is associated with slow movement.

As a reminder, lower RDP values mean that IPM performs better than ALM and vice versa. Judging the performance of both we can easily see that ALM performs best with fast movement (1) and IPM with slow movement (2). So IPM is superior for less mobile nodes.

The last figure shows the results for the link stress measurement:

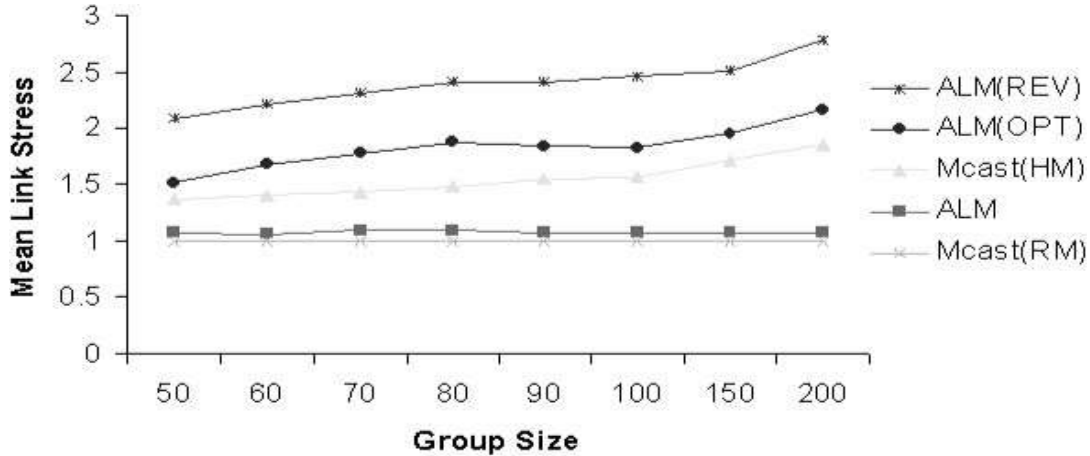


Figure 4: "Mean link stress for mobile hosts" [f4]

Here, the x-axis shows the size of the groups and the y-axis the mean link stress. ALM(REV) means ALM reverse tunnelling, ALM(OPT) is ALM optimized routing, Mcast(HM) is IPM home subscription, Mcast(RM) IPM remote subscription. This figure shows interesting results: The link stress is always 1 for IPM with remote subscription, which is almost the same value as with generic ALM. The greater values for IPM with home subscription are caused by reverse tunnelling needed for the packets to reach the node. Also, we can see that ALM with reverse tunnelling is a worst case scenario for moving nodes and implementing optimized routing shows better results.

In general, we can say that ALM causes an overhead and that ALM packets traverse a link 1.7 times more than IPM.

Concluding the above results, ALM shows no benefit over IPM. The concerns were confirmed. Although IPM showed no advantage with increased mobility, ALM suffered from increased packet loss with high mobile nodes. Regarding RDP, IPM showed far better performance with low mobility – up to a factor of 4-5 –. This value decreased with faster movement to a factor of 2. This metric is heavily depending on user behaviour, localized movement showed a smaller gain for IPM. Link stress with ALM was about 1.7 times higher than with IPM and generally increased with group size. IPM still outperforms ALM in all aspects. Although no protocol support for ALM is needed, it is questionable if it will ever

work. The authors of [9] therefore suggest a smarter and hybrid approach with IPM providing support for intra-domain and ALM for inter-domain routing.

Next we will see if using peer-to-peer overlays for ALM proved more beneficial for mobile networks.

3.2 ALM using peer-to-peer overlays

As mentioned before, [10] used CAN and Pastry for peer-to-peer overlays and implemented each with flooding and tree-building for multicasting. The simulation model was set up using a packet-level event simulator on five network topologies with 5050 routers and 80.000 end nodes. On this setup, two sets of tests were run. One with one single large group containing all end nodes, the second with 1500 groups.

For measurement, the same criteria as in [9] were used. The relative delay penalty RDP, which was computed by dividing both the maximum and average delay on ALM by maximum and average delay on IPM, link stress and duplicates (which is another metric for robustness). Both implementations, CAN and Pastry, were used, each with flooding and tree-based multicast. First we will have a look at the results for CAN, then at those for Pastry. We will also see which results enabling different parameters will show.

When using CAN as implementation for building a structured peer-to-peer overlay for ALM, enabling the landmark-based assignment showed largest improvement for average delay. Using the flooding algorithm for multicasting showed, regarding the RDP, that it is independent from the routing table size. For link stress, it showed that the landmark-based assignment lowered the amount of packets to traverse the network. Notable was, that when all 80.000 nodes joined a group the link stress was far higher than when a message was sent to all 80.000 nodes. The impact of duplicates (the robustness) on the performance was neglectable.

The second set of tests was run using PASTRY as implementation for multicasting. In this case, two optimizations were used: the topology-aware nodeID assignment (*TOP*) and topology-aware routing table construction (*TART*). When using the flooding algorithm, it showed that the RDP value was lowest when combining both TOP and TART, which reduced the delay penalty by up to 60 %. Using these optimizations, also the link stress dropped by 30 %.

Another interesting outcome was that increasing the number of bits b of the nodeID used by the routing algorithm to find the next node, really increased the overall performance. Unfortunately, it also increased the number of duplicates enormously up to a factor of 1000. This is caused by the no longer up-to-date routing tables in the nodes, though which can be repaired at relatively low cost.

When using a tree-based approach for multicasting with PASTRY, RDP showed the same values. Though this does not seem to be an improvement, the link stress was still even lower than with the flooding approach.

The general outcome showed that when peer-to-peer overlays are used for ALM, two combinations were superior in terms of robustness. Namely these were PASTRY with flooding and topology-aware nodeID assignment for a single large group and PASTRY with tree-based algorithm and topology-aware routing table construction for many groups. These should be implemented in accordance to the topology and amount of mobility in the underlying network.

4. Conclusions

The results from [9] and [10] showed that Application-level multicast itself is not a solution for problems in mobile networks. Especially with fast moving nodes, which is typical for a modern mobile environment, it performs even worse than IP multicast. Therefore, the authors of [9] suggest a hybrid approach where both, ALM and IPM are used. ALM for inter-domain and IPM for intra-domain routing.

Even when applying structured peer-to-peer overlays to implement ALM, as proposed in [10], where ALM's performance is much better, it still can not beat IPM in terms of performance. Also, when using peer-to-peer overlays, only tree-building algorithms for multicasting are beneficial.

5. Personal thoughts

The authors of [9] presented a very good overview and a deep insight into the topic of Application-level multicast. They followed their idea of the outcome clearly and used a thoughtful way to evaluate their ideas using a well set-up simulation. A drawback was the wrong-numbered figures which were referred to erroneously with false numbers. Also it is unclear why the authors put the term "mobile IPv6" in their work's title. Besides a short introduction to Mobile IPv6 they did not really refer to the topic, although they claim to analyse "the implications of running multicast in a mobile, wireless network using Mobile IPv6" ([9], p. 58). This whole part of the work was not done clearly.

Besides not clearly stating which thoughts lead to choosing only CAN and PASTRY as representatives of peer-to-peer overlays, the authors of [10] released a good work. Only drawback was the too huge amount of figures and a maybe too in-depth coverage of the specific topic.

6. References

- [1] T. Harrison, C. Williamson, W. Mackrell and R. Bunt, "Mobile Multicast (MOM) Protocol: Multicast support for Mobile Hosts", MOBICOM, Budapest, Hungary, Sep. 1997.

- [2] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level multicast using content-addressable networks", in Proc of NGC, Nov. 2001.
- [3] I. Stoica, R. Morris, d. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer look up service for internet applications", in Proc of ACM SIGCOMM, San Diego, California, Aug. 2001.
- [4] A. Rowstrom and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", in Proc of Middleware, Nov. 2001.
- [5] B. Zhao, J. Kubiawicz, and A. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing" U.C. Berkley, Tech. Rep. UCB//CSD-01-1141, Apr. 2001.
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network" in Proc. of ACM SIGCOMM, Aug. 2001.
- [7] A. Rowstrom, A.-M. Kermarrec, M. Castro and P. Druschel, "Scribe: The design of a large-scale event notification infrastructure", in Proc of NGC, Nov. 2001.
- [8] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstrom, "Scribe: A large-scale and decentralized application-level multicast infrastructure", IEE JSAC, vol. 20, no. 8, Oct. 2002.
- [9] A. Garyfalos, K. Almeroth, J. Finney, "A Comparison of Network and Application Layer Multicast for Mobile Ipv6 Networks", MSWiM'03 San Diego 2003
- [10] M. Castro, M.B. Jones, A.-M. Kermarrec, A. Rowstrom, M. Theimer, H. Wang, and A. Wolman, "An evaluation of Scalable Application-level Multicast Built using Peer-to-peer Overlays", IEEE Infocom 2003

Figures:

[f2],[f3],[f4] taken from [9]