Avoiding Unnecessary Congestion Responses in Layered Multicast with Heterogeneous Receivers

Chamil Kulatunga and Gorry Fairhurst
Email: {chamil, gorry}@erg.abdn.ac.uk
Electronics Research Group, Department of Engineering, University of Aberdeen
United Kingdom, AB24 3UE

Abstract - Receiver-driven layered multicast has been designed to support scalable multimedia applications and very recently used to support bulk file transfers. Layered multicast congestion control assumes that all the receivers behind a bottleneck are equally able to subscribe to a particular number of cumulative layers. This paper explores the requirement placed on heterogeneous receivers behind a bottleneck router link. In such a scenario, leaving a layer by a less capable receiver (as a result of a congestion response) leads to resource underutilization in multicasting. Three approaches to a solution to the problem are discussed together with their limitations.

Index Terms - scalable content, layered multicast, traffic prioritization, shared feedback announcements, explicit congestion notification

I. INTRODUCTION

The risks of non-responsive flows (i.e., flows that do not respond to congestion) in the Internet has been emphasised by Internet researchers over a decade [1]. The effect of multicast without congestion control will fuel this problem as multicast applications become more popular. The IETF Reliable Multicast Transport (RMT) working group started a process called Application Layer Framing (ALF) that implements functionalities that can improve multicast over UDP. This work is in parallel to the emergence of more general but feature negotiable transport protocols for unicast like DCCP [2].

In devising end-to-end multicast congestion control, layered multicast has become recognised as a scalable mechanism. A requirement to have scalable applications and complexity to achieve TCP friendliness have paved a path to sender-driven single-rate multicast congestion control protocols.

The proposed mechanisms for layered multicast congestion control assume that all the receivers behind a bottleneck can subscribe to a common set of layers. Receivers behind a router bottleneck can however be heterogeneous and the location of the bottleneck also can be at any link along the multicast tree. Therefore leaving layers by less capable receivers is unnecessary when a receiver with the highest capability responds to the congestion by leaving its highest layer reducing the total throughput by half. This paper analysis this phenomenon, which leads to a more optimised layered multicast protocol. Approaches for a solution are discussed but no solution is devised in detail in this paper.

The paper is organised as follows. Section I provides an introduction. Section II discusses multicast in a satellite environment with a scenario where the problem can occur. Section III describes layered multicasting with scalable content distribution. Section IV describes the layered multicast congestion control mechanism with a brief introduction to other multicast congestion control schemes. Section V discusses the problem in detail when the assumption of equally capable receivers is not valid over a bottleneck. Both the static and dynamic layering mechanisms are analysed under this section. Section VI presents simulation results that demonstrate that the problem leads to resource underutilization for Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) protocol using ns2 simulations [3] [4]. Section VII discusses possible solutions from three different approaches. Finally Section VIII ends with the conclusions.

II. SATELLITE MULTICAST

The broadcast nature of satellite networks makes them particularly suited to multicasting as an affordable and user-oriented service. Satellite networks are being developed with a satellite return channel [5] for use in SOHO scenario and to extend communication coverage for rural areas. In DVB-S2/RCS [6] satellite systems, end terminals are integrated with low cost wireless and wireline access technologies as a backhaul connection. In these scenarios, as shown in Fig. 1, one terminal that support return channel, RCST is used as the access point to the satellite and other access technologies are integrated to extend the services beyond that point. That allows a share of the cost of the expensive RCST terminals and to maximise the use of the return channel.

This work is carried out as part of the project “Satellite-based Secure Multicast Employing Hybrid Reliability” funded by EPSRC.
The end users behind a RCST might use differing types of terminals such as PDAs, laptops, desktops or mobile phones and can be used over different access technologies e.g. WiFi, WiMax, LAN, even dial-ups. Receiver capabilities in combination with last-mile bandwidth behind a RCST can vary from user to user. Therefore the problem discussed in this paper is valid in satellite networks as well.

III. SCALABLE CONTENT DISTRIBUTION AND LAYERED MULTICAST

**Scalable Content Distribution:** Scalable content delivery may be primarily used to provide adaptive QoS for unicast multimedia applications over dynamic networks. Due to different network capabilities and traffic patterns applications tend to be developed with scalable content coding, so that traffic can be adapted to the available network bandwidth while, preserving a deterministic quality. This design allows the application to become more flexible and allows it to flow through bottlenecks in the network while optimising network resources and user performance. The concept of scalable content distribution first came into the Internet as Scalable Video Coding (SVC) [7]. In SVC, quality, special (resolution) and temporal (frame rate) scalabilities are achieved.

In multicasting, receiver heterogeneity and congestion control are addressed by scalable content distribution. The scalable content distribution has started to move beyond multimedia applications. The Asynchronous Layered Coding (ALC) framework [8] provides scalable reliability for file transfers.

Receiver-driven layered multicast has emerged as a potential candidate for bulk multicast applications that addresses receiver heterogeneity and congestion control. More importantly receiver-driven layered multicast has successfully addressed the problem of scalability and is completely independent of the number of receivers in the multicast session.

The ALC protocol has been standardised as the Layered Coding Transport (LCT) building block [9] for session management in the IETF RMT working group. Scalable applications can transport data using the framework as layered objects. The framework has provision to integrate with building blocks to support extended functionalities that fulfil an end-to-end service. Functions include reliability, congestion control and security. The RMT working group has focussed on bulk data transfers where multicasting will be highly effective since the probability of having multiple receivers in a close proximity is very high for such services. This service is useful for applications such as automated software updates.

The FLUTE protocol [10] has been designed for unidirectional file transfers based on the ALC framework, with FEC as the reliability mechanism [11]. Within the project “Satellite-based Secure Multicast Employing Hybrid Reliability”, FLUTE is expected to be extended to offer congestion control and security.

Layered multicast congestion control is a congestion control building block for ALC. In layered multicast, the sender transmits layered objects using multiple multicast groups. The receiver joins layers independently as required. In congestion, the receiver leaves layers as a response to the congestion. Then the receiver re-joins with enhancement layers, as bandwidth is available again along the path from the sender to the receiver.

**Heterogeneity:** According to the end-to-end path characteristics and receiver performance, a receiver can join with an appropriate number of cumulative layers in a multicast session. This is determined by the network bandwidth, local ISP’s resource limitations or level of subscription, receiver’s processing power and memory space etc. Therefore the number of layers to which any receiver can subscribe may not be completely governed by the congestion.

**Scalability:** Scalability is achieved by granting total autonomy to the receivers to monitor and acquire data appropriately from the set of multicast groups. Therefore a feedback channel is not offered.

IV. RECEIVER-DRIVEN LAYERED MULTICAST

**CONGESTION CONTROL**

Multicast congestion control may be provided by two approaches, router-support and end-to-end [12]. In router-support approaches such as MTCP [13] and RMTTP [14], a hierarchical tree structure is used that performs local retransmissions and ACK aggregation along the sub-trees. This requires upgrading existing network infrastructure and a processing overhead due to complex functionalities at the routers. This approach has not been standardised. Lightweight support from routers for a hybrid mechanism with an end-to-end technique such as Explicit Congestion Notification (ECN) marking in TCP is not completely ruled out.

The IETF RMT working group’s main focus has been to standardise end-to-end multicast congestion control protocols. Two major divisions are in progress. One is the sender-driven multicast congestion control and other one is the receiver-driven layered multicast congestion control, which is the basis for this paper. In a sender-driven approach the sender adapts its rate according to the receiver having the smallest bandwidth availability in the group. Sender adaptation to congestion is achieved in PGMCC [15] with TCP like window-based and in TFMCC [16] with TFRC rate based controls. The major drawback of the approach is all receivers need to depend on a single rate that is the slowest receiver of the group, which is dynamically selected. Scalability has been achieved through a random timer based NACK suppression mechanism. Evolution of sender-based multicast congestion control is important when non-scalable applications are used in multicast. It is TCP friendly compared to the layered multicast congestion control.

Receiver-driven layered multicast addresses receiver heterogeneity. In DSG [17] video applications transmit in three different multicast groups with differing bandwidths and qualities. Then receivers join with one suitable group according to bandwidth capabilities. In this approach, scalable applications were not a requirement but the data were replicated over several multicast groups reducing the benefit of multicasting.

In contrast cumulative layering is where a receiver can join layers consecutively to enhance the performance of the reception. RLM [18] and PLM [19] were first proposed with static layers, where each layer carried data at a constant rate. Layer rates are exponentially spaced to mimic TCP
multiplicative decrease behaviour for congestion responses. When there is a packet loss, the highest layer was dropped. Each receiver calculates the target rate based on the TCP throughput equation and joins with the next consecutive cumulative layer if it does not violate the achievable target rate. A join to a new layer is done using an Internet Group Management Protocol (IGMP) join message and once received, the multicast path is set with graft messages causing traffic to flow into the network.

**Synchronization Points:** When any receiver joins a new layer if there is any other receiver in that network who can get that data rate should join with the same number of layers without being underutilised. When a receiver joins it may result in the network becoming congested and unnecessarily other receivers could leave their highest layer. To avoid this problem, the sender initiated synchronization points, which are embedded in the data streams, so that all receivers behind a bottleneck join with a layer at the same time.

In multicast, IGMP informs routers to join/leave receivers. Accordingly the router maintains a path from the sender to the sub-network as part of the multicast tree. Layered multicast requires consideration of IGMP since performance depends on leave latency. IGMP leave latency played a major role in the congestion response delay and complex dynamic layers was proposed instead of static layers. With the standardization of IGMPv3 [20] the problem of leave latency is solved subject to its widespread deployment.

As a result of congestion the receivers leave from the highest layer first. In IGMPv2 time to get a leave message from a receiver and prune the route from the multicast tree is as high as 3s. Therefore traffic continues to flow into the network during that time, without responding to congestion. When the join/leave is via a satellite link, the additional delay further aggravates this problem.

**Bandwidth Probing:** Failed bandwidth-increase attempts also can congest the network due to the IGMP leave latency. Therefore before joining a group, the sender probes the path to make the receiver aware about the status of the path. This is done using a burst of traffic that mimics the next layer in RLM or using a packet-pair technique in PLM. In future network support for protocol mechanism like Quick Start [21] may allow this probing to be done more accurately but currently such methods are only defined for unicast.

To address IGMP leave latency, layered multicast was changed to dynamic layering where the rate at each layer was changed with time. FLID-DL [22] uses stepwise wave shapes and Wave and Equation-Based Rate Control (WEBRC) [23] uses continuous exponentially decaying wave shapes. With dynamic layering, instead of leaving layers to respond to congestion, the receiver remains subscribed to a number of layers without subscribing to any new layer to respond to congestion. Then automatically the rate from the sender decreases with time. A receiver leaves layers when the rate at a wave becomes zero.

V. OPTIMISING LAYERED MULTICAST

In current layered multicast congestion control mechanisms, receivers monitor the lost or marked packets as an indication of congestion. If there are lost or marked packets a receiver leaves the highest layer it has subscribed. If exponentially incremented cumulative layers are used, this mimics the TCP multiplicative decrease congestion response behaviour.

In the current best-effort Internet packets are dropped or marked randomly due to queue overflows or RED marking. Random losses can happen irrespective of the flow in both RED and Drop-tail queuing disciplines. Therefore in layered multicast, packets can be dropped randomly from any layer at any time by any amount due to congestion. Probably lower layers might show packet losses while the highest layer does not. Therefore the receivers should monitor for lost and marked packets in all subscribed layers to get an indication of congestion. All losses are treated equal, it doesn’t matter which layer is having packet losses, the receiver should leave from the highest layer it has already subscribed. A problem arises in this concept when the receivers with different capabilities share a common bottleneck. The problem is described below.

According to Fig. 2, receivers A and B can join with all four layers and receiver C can join with three layers, but the receivers D and E can join only with the lower two layers.

**Case 1 (Behind a Common Bottleneck):** Assume that due to some other flows, unicast or multicast, congestion developed in between nodes 1 and 2. We assume that packets are dropped from Layer 2. Then all the receivers experience the congestion and leave their highest layer. A and B leaving from layer 3 might have healed the congestion and they are receiving lower three layers. In such a situation C, D, and E leaving from any layer leads to unnecessary resource underutilisation.

**Case 2 (Behind Separate Bottlenecks):** If we assume there are two sources of congestion at the same time in between nodes 2 and 3 as well as 4 and 5. Receivers A, B, D, and E will have congestion indications. In this case A and B as well as D and E should leave from their highest layers, even though they are two different layers, in order to heal both congestion spots.

Ideally this problem can be avoided if packets are always dropped from the highest layer of the multicast session over a bottleneck. The problem occurs equally in static and dynamic layering techniques. With static layering additional IGMP leave/join messages due to unnecessary congestion responses can be very high. By avoiding those unnecessary congestion responses, expensive satellite return channel resources can be saved significantly.
VI. SIMULATION RESULTS

The topology shown in Fig. 3 is simulated with ns2 implementations of FLID-DL [4]. CBR traffic was applied to congest the link common to all receivers at time 100s and stopped at 105s. Congestion responses of the three receivers were observed. Tail drop queue with 20 packets is used for the congested link.

![UDP/CBR Traffic at 2.5Mbps for 5s](image)

Fig.3. Simulated topology with FLID-DL

Dynamic Layering: Rate at each layer is varying with time in dynamic layering. There is a set of identical time varying layers called “waves” which exist at a particular time with differing rates at each wave. Rate can be changed stepwise, as in FLID-DL, or continuously, as in WEBRC. We have simulated the problem with FLID-DL and this should be equally applicable to WEBRC as well. Exponentially distributed layers with multiplication factor of 1.3 were used. Base layer rate was 24 kbps.

According to Fig. 4, all three receivers start to respond to congestion at 100s by leaving layers. Receiver A drops its subscribed number of layers from 20 to 14 towards the end of the congestion at 105s. Therefore 14 lower layers get through the bottleneck during the congestion. In this scenario, B and C have no requirement to leave any layer below 14. But according to the results, B and C leave layers up to 11 and 8 respectively.

To improve fairness with TCP, layered multicast protocols provide frequent synchronisation points for lower layers than upper layers. Hence lower layers have a higher ability to improve its dropped rate quickly. According to Fig. 4, C reaches back to its original status quicker than A. The problem described in the text should equally be valid for both types of layer techniques.

VII. DIRECTIONS FOR A SOLUTION

Solutions to the problem can be addressed with or without underlying network support. Three different approaches are described below with their limitations and dependencies. We analyse the solutions only for static layering. It shows difficulties of finding a solution with dynamic layering than static layering. Since IGMPv3 avoids the requirement of dynamic layering the authors believe that there is no advantage of considering it for dynamic layers. Directions of the solutions we discussed are,

A. Layer prioritization
B. Router-support mechanisms
C. End-to-end (receiver coordinated) mechanisms

A. Layer Prioritization

Layer prioritization can be achieved end-to-end or locally. End-to-end approach is the QoS architecture for the Internet we need to have over the underlined network. But in local prioritisation, traffic will be prioritised over a single or few hops along the path.

Priority-based Networks: If the underline network supports traffic prioritization then the lowest layer can be assigned priority and priority can be degraded as the layers. This is an ideal network environment for layered multicast congestion control. The application can send less important enhancement information over higher layers so that this is only sent when the network has resources. Then highest layer packets are dropped first and other layers will not loose any packets in congestion. The receivers can monitor congestion only at the highest layer. If losses are observed, then the highest layer will be dropped. Practical reasons such as a scalable management architecture for admission control and billing (DiffServ) or non-scalability at the routers to maintain per flow states (RSVP), QoS enabled end-to-end networks can not be expected in the near future.

Local prioritization: Layers can be prioritised at a single link or a network segment in a service provider’s administrative domain using priority based queuing where congestion is most likely. In satellite networks this can be possible since traffic is traversing a limited number of hops and often only a single service provider. But in terrestrial networks this is hard to deploy. The bottleneck can be any where along the data path that may comprise several tens of hops and several ISPs. Topology to the end-user is completely unaware and the inter-domain provisioning
remains as a major complexity for ISPs in QoS consideration.

B. Router-support Mechanisms

In router-supported approaches like ECN [24] packet marking, used for unicast can be considered as a mechanism to improve multicast congestion control. The indication for congestion in advance in conjunction with active queue management needed to be modified for layered multicast. A good approach is the marking packets, we call Layered Marking (LM), in the highest layer in a layered multicast session when a packet is dropped or marked with ECN at any layer below it. In Fig. 5 there are four layers in the multicast session and a packet is dropped from layer 1. At the same time, the router marks a packet at layer 3 with the congestion indication. In this situation the receivers who have subscribed only up to layer 2 will experience packet losses or marked packets but in the absence of LM the receiver does not leave any layer. But receivers who have subscribed to all four layers will get packet losses at layer 1 as well as LM marked packets. As a response to the congestion, the receivers who gets LM packet will immediately leave layer 3, which is the highest layer along the bottleneck and reducing the congestion.

In this approach the receiver functionally is simple to implement and the layered multicast sender operation is not changed.

**Router functionality:** Additional router functionality is the major part of the approach. In packet drop or ECN marking the router needs to identify a session if it belongs to a layered multicast session. Then wait and identify a packet that belongs to the highest layer of that session through the particular link. Then that packet is marked with LM. The identification of the session needs further modifications to the architecture as the session identification of layered multicast is above the transport layer in the ALC framework. Therefore mapping of the session and layer identifications to the IP header is needed. This requires changes to the underlying network architecture. Scalability issues come when use with large number of flows. A lightweight mechanism cannot be ruled out and open for further research as layering cannot be avoided in multicasting. Using this approach the scalability of receiver-driven layered multicast congestion control can be preserved.

C. Receiver Coordinated Mechanisms

We propose a coordination mechanism among the receivers using a controlled feedback channel. Addressing Case 1 and Case 2 at the same time is the biggest challenge to face in the solution of the problem. In addressing this situation we assume that we are working on the best-effort Internet and no traffic prioritisation is deployed.

If any receiver has subscribed to the maximum number of possible layers and is experiencing congestion at any layer they should leave the highest layer immediately. It should then announce its congestion response to others to avoid unnecessary demotions if they are under the same bottleneck such as in Case 1. If they are not under the same bottleneck weaker receivers should not obey that announcement and leave their highest layer to respond to the congestion as in Case 2.

A multicast feedback channel can be used similar to TFMCC feedback round that goes for 4-6 RTTs. To address Case 1, access priorities for feedback channel can be assigned according to the number of layers they have subscribed. Access method for the announcement channel can be time division, as shown in Fig. 6. Receivers, who have subscribed to the maximum number of layer, need to announce congestion responses in the first Time Slot (TS). Then the receivers, who have subscribed to one layer lesser, need to wait for the second TS to announce their congestion responses. At the same time there should be adequate information in the feedback report so that others who are experiencing congestion are able to get knowledge about whether it is a common bottleneck or not. Therefore knowing the congestion bottleneck in the multicast tree should also be addressed.

**REFERENCES**


