Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Networks 50 (2006) 80–105

Computer Networks

www.elsevier.com/locate/comnet

AQoSM: Scalable QoS multicast provisioning in Diff-Serv networks ☆

Jun-Hong Cui [a,*], Li Lao [b], Michalis Faloutsos [c], Mario Gerla [b]

[a] Computer Science & Engineering Department, University of Connecticut, Storrs, CT, United States
[b] Computer Science Department, University of California, Los Angeles, CA, United States
[c] Computer Science & Engineering Department, University of California, Riverside, CA, United States

Received 8 November 2004; received in revised form 17 February 2005; accepted 29 March 2005
Available online 31 May 2005

Responsible Editor: S. Palazzo

Abstract

The deployment of IP multicast support is impeded by several factors among which are the state scalability problem, the cumbersome management and routing, and the difficulty of supporting QoS. In this paper, we propose an architecture, called Aggregated QoS Multicast (AQoSM), to provide scalable and efficient QoS multicast in Diff-Serv networks. The key idea of AQoSM is to separate the concept of groups from the concept of distribution tree by "mapping" many groups to one distribution tree. In this way, multicast groups can now be routed and rerouted very quickly by assigning different labels (e.g., tree IDs) to the packets. Therefore, we can have load-balancing and dynamic rerouting to meet QoS requirements. In addition, the aggregation of groups on fewer trees leads to routing state reduction and less tree management overhead. Thus, AQoSM enables multicast to be seamlessly integrated into Diff-Serv without violating the design principle of Diff-Serv of keeping network core "QoS stateless" and without sacrificing the efficiency of multicast. Finally, efficient resource utilization and strong QoS support can be achieved through statistical multiplexing at the level of aggregated trees. We design a detailed MPLS-based AQoSM protocol with efficient admission control and MPLS multicast tree management. By simulation studies, we show that our protocol achieves significant multicast state reduction (up to 82%) and tree maintenance overhead reduction (up to 86%) with modest (12%) bandwidth overhead. It also reduces the blocking ratio of user requests with strong QoS requirements due to its load balancing and statistical multiplexing capabilities.

☆ This material is based upon work supported by the National Science Foundation under Grant No. CNS-0435515, CNS-0435230, ANIR-9985195 (CAREER award), and IDM-0208950.
* Corresponding author. Tel.: +1 860 486 8951.
E-mail addresses: jcui@cse.uconn.edu (J.-H. Cui), llao@cs.ucla.edu (L. Lao), michalis@cs.ucr.edu (M. Faloutsos), gerla@cs.ucla.edu (M. Gerla).

1389-1286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2005.03.003

Keywords: Multicast; QoS; State scalability; Diff-Serv; MPLS

1. Introduction

Real-time multicast applications need mechanisms to support (if not guarantee) QoS. This is especially compelling in the case of commercial applications: customers will not pay for a service unless it is reliable and offered at a satisfactory level of quality. From the perspective of network service providers, QoS multicast provisioning requires three mechanisms: (a) discovery of available resources, (b) maintenance of the available resources, and (c) quick recovery from failures. These requirements translate to the following functions: (a) QoS-aware routing, (b) call admission and load-balancing, and (c) fault-tolerant routing.

The current multicast architecture does not handle QoS efficiently due to the following issues. First, multicast routing state does not scale well: routers need to keep state per group and in some protocols per group/source. Large numbers of groups result in large amount of state to be maintained at routers, which translates into large memory requirements and slow packet forwarding. Second, routing is cumbersome. A tree is associated with a single group. Creating and maintaining a multicast tree per group is time and resource consuming, especially if we account for QoS constraints. As a result, a congested link or a node failure leads to tearing down and rebuilding large parts of the tree. Finally, load-balancing and rerouting a tree is not an option.

Though most research papers on QoS multicast focus on solving a theoretical QoS-constrained multicast routing problem, there have been several more pragmatic efforts to bring QoS into the existing IP multicast architecture, such as YAM [9], QoSMIC [18], QMRP [10], RIMQoS [22], QoS extension to CBT [25], and PIM-SM QoS extension [5]. However, we find that the problem can only be addressed by an architecture that will address all three QoS aspects listed earlier, namely QoS aware routing, admission control, and fault tolerance. The proposed protocols address only one of the requirements, usually the QoS aware routing. Furthermore, they all use per-flow state and continue to be plagued from the state scalability problem. Today people are backing away from micro-flow-based QoS architecture, for example, the Integrated Services architecture (IntServ) [8], and are moving towards the aggregated-flow-based solutions, such as Differentiated Services (Diff-Serv) [6] architecture and the Multiprotocol Label Switching (MPLS) technology [39]. The argument behind this choice is simple: the per-flow reservation and data packet handling required by IntServ lead to scalability problems at network core routers. Incorporating the per-flow state requirement and traffic management of multicast in a per-class architecture, such as a Diff-Serv or MPLS network, does not solve the state scalability problem, since each router still needs to maintain separate states for individual multicast groups that pass through it. To improve multicast state scalability, several mechanisms have been proposed, such as [50,44,12,49,38,7], etc. Though promising, these efforts are typically not concerned with QoS. Recently, there are also some works which target at scalable QoS multicast provisioning [28,48,45]. However, they either do not scale to large groups or sacrifice the efficiency of multicast.

In this paper, we propose an architecture, called Aggregated QoS Multicast (AQoSM), to provide scalable QoS multicast that addresses the issue of QoS and routing in a unified and comprehensive way. Our architecture intends to be employed in a Diff-Serv-supported transit domain and its use is transparent outside the domain or to the application layer. AQoSM uses the concept of aggregated multicast [21], in which the key innovation is the decoupling of group and distribution tree concepts. Many groups can be multiplexed on a single tree. More importantly, a group can be switched easily between distribution trees. This simple feature leads to a proliferation of new properties and advantages. First, the creation and management of trees become more efficient. We can create trees on-demand and route a group very quickly. Second, group *rerouting* becomes a viable