Call Admission Multicast Protocol (CAMP) for End-to-end Quality-of-Service *

Elena Pagani, Gian Paolo Rossi
Dip. di Scienze dell'Informazione - Università degli Studi di Milano

Extended Abstract

A great deal of work in the field of the Internet deployment has been recently devoted to the caracterization of approaches to guarantee *Quality-of-Service* (QoS). The needs for QoS support come from the large diffusion of distributed multimedia applications, such as the web-based applications. Those applications require a data transmission service better than the best effort service currently offered by IP, that does not provide any guarantee in terms of reliable and timing data delivery. The particular service to be provided depends on the requesting applications [6, 2]. It could be for instance defined in terms of high reception rate, low end-to-end transmission delay or low packet loss probability. The vast range of QoS requirements makes difficult to devise mechanisms that are general enough to guarantee that those heterogeneous requirements are fulfilled. As the consequence, in spite of the large amount of proposals in the literature, no standard is so far emerged and many problems are still far from being solved. Moreover, the most part of the multimedia applications are characterized by a one-to-many or a many-to-many communication pattern. Hence, they demand for network protocols supporting *multicast* [5].

The problem of providing Quality-of-Service (QoS) concerns all the architectural layers [10]. The proposals appeared in the literature usually focus on the network and data-link layers, specifically, on routing, resource reservation and packet scheduling functionalities [3, 4, 1]. However, those proposals require the upgrading of all the network elements (hosts and routers) constituting the Internet. In the short term, the support to QoS can be guaranteed by working at the *end-to-end layer*, that is, by designing call-admission and signaling protocols that exploit the characteristics of the existing network protocols to find out whether and where enough resources are available to accommodate the QoS traffic.

In this frame, we propose the *Call-Admission Multicast Protocol* (CAMP). We hypothesize that the underlying layers are QoS-unaware. CAMP only needs that a multicast routing protocol is available (e.g., DVMRP), and that all the routers in the network are configurable to use the priority packet scheduling policy. When an application issues a request for QoS, it informs CAMP about the profile of the traffic it will generate, and the destinations of that traffic. CAMP uses this information to test if the network can guarantee the requested quality. The test consists in sending `probe` packets to the destinations, so that the generated probing traffic has the same profile as the forthcoming data traffic. Best effort packets have lower priority than the `probe` packets, that have lower priority than the data packets of the already established QoS connections. This way, the probing traffic does not affect the existing QoS flows. On the other hand, it drains available bandwidth for new QoS flows at the expenses of the best effort traffic. At the receiver side, the CAMP entity evaluates the quality of the received probing traffic. CAMP exploits the services of the RTCP libraries [11]: the sender diffuses its own traffic profile using the RTCP reports. Each receiver r matches that profile against that of the received probing traffic and decides whether to accept the data flow or not. r informs the sender about its decision with its own RTCP reports. If the service is accepted, then the sender switches from the transmission of `probe` packets to the transmission of the data packets generated by the source application. The priority policy ensures a sort of resource reservation: it guarantees that the bandwidth that was available to a connection during its probing phase cannot be later stolen to QoS packets by either `probe` or best effort packets. A receiver that refuses the service, prunes itself from the multicast routing tree. In figure 1 we show the system architecture and a layout of the CAMP communication pattern.

*This work was supported by the MURST under Contract no.9809321920 "Transport of multicast packets with QoS guarantees".

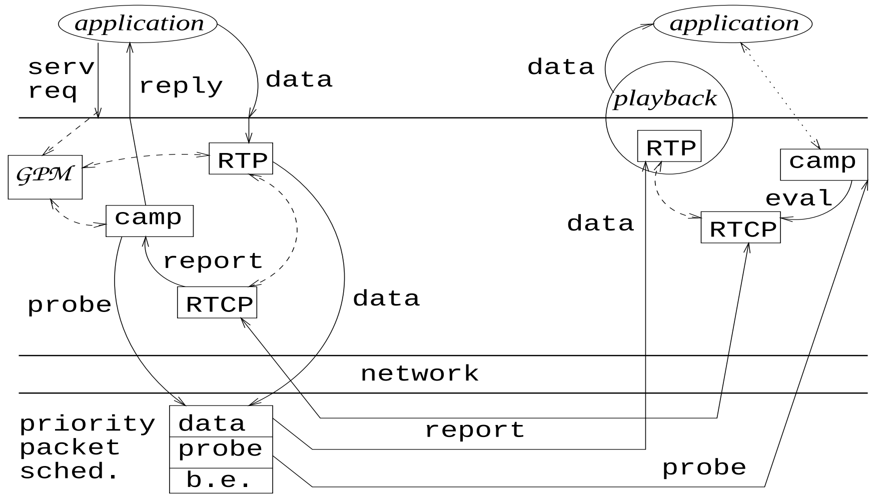

Figure 1: Layout of the CAMP architecture and communication pattern.

This mechanism can be used to provide QoS in *Differentiated-Service* domains [3], because it naturally performs flows aggregation. Some problems have to be faced when adopting this approach for multicast communications. The QoS-unawareness of the intermediate routers forces to adopt source-based trees as the multicast routing infrastructure, rather than shared trees. In the latter case, the intermediate routers are not able to forward different subsets of flows to different receivers. A receiver r belonging to a shared tree must accept all the data traffic forwarded along the tree. If there are not enough resources available, r must leave the tree, thus not receiving any flow. By contrast, with source-based trees a receiver joins the trees of only the sources it is interested in and whose traffic can be managed by the network.

The sender *must* receive the reports from *all* the receivers, before switching to the data transmission. Consider the example shown in figure 2. If the sender receives positive reports from the receivers in the

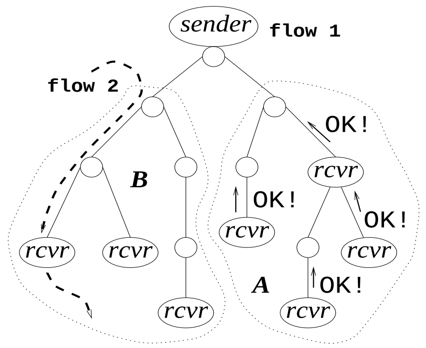

Figure 2: Example of concurrent flows.

subtree A, it knows that destinations exist for the data flow. If the sender switches to the data transmission before receiving the reports from the receivers in the subtree B, other ongoing QoS flows could be negatively affected. Indeed, other flows could traverse the B's branches, that do not leave enough bandwidth for the sender flow. As the consequence, the receivers in B would eventually send negative reports and prune to refuse the new flow. After the switching, the data packets sent to B are aggregated to those of the other ongoing flows. Since resources are scarce for the aggregated traffic, packets can be dropped, belonging to whichever of the aggregated flows.

To know that all the reports have been received, the source must know the destination group membership. This can be achieved by adopting a membership protocol. For instance, a directory service can be used by the traffic sources to announce their intention of transmitting and the transmission start time. The interested receivers must subscribe to the source and join the distribution tree before the time at which CAMP is run.

An alternative is to exploit the characteristics of RTCP. Each receiver periodically sends its own report to both all the other receivers and all the sources it knows. A node learns of a source existence by receiving a packet from that source. Let *rtd* be an upper bound on the round-trip delay between the source and the farest destination; let p_{RTCP} be the upper bound on the period with which a receiver sends its RTCP reports. If the probing phase lasts at least $rtd + p_{RTCP}$, the source receives at least one report from each receiver, thus learning the receiver group membership. As the consequence, it can then extend the probing until a decision has been received from each receiver. Differently from the directory service, this approach is more flexible but it requires that some knowledge is available about the network topology.

The need of obtaining all the decisions requires as well that report loss is prevented, for instance by having RTCP that uses the TCP services.

We have implemented the architecture shown in figure 1, in the frame of the NS-2 simulation tool [7]. We take as the multicast routing protocol DVMRP. We are performing the preliminary experiments to compare the achieved QoS by using either the described end-to-end signaling mechanism or a QoS routing protocol such as QoSMIC [8], that realizes as well call-admission and resource reservation functionalities. We are evaluating how CAMP behaves when several QoS flows compete among them and with the best effort traffic for the available bandwidth. Some results will be discussed in the final version of this paper.

Concurrently, we are devising a mechanism to overcome the CAMP limitations discussed above. This mechanism consists in instantiating CAMP *proxies* in the in-tree routers. Proxies can be used to hide the destination membership to the sources, by exploiting the knowledge of the tree structure in the routers. They can also support heterogeneous receivers belonging to the same distribution tree and having different resource availability. A proxy can record the (possible different) subsets of flows to forward over the different subtrees. Similarly, the mixing and translation functionalities of RTP could be embedded into the proxies.

In the medium term, the proxy approach can guarantee a more flexible support to the end-to-end QoS than the pure CAMP; at the same time it does not require great changes to the existing routers. We plan to implement the proxies in our simulation environment. In the near future, we will be able to perform experiments with the described architecture in the frame of a testbed that we are currently deploying over the departmental network [9].

References

[1] Aurrecoechea C., Campbell A. T., Hauw L. "A Survey of QoS Architectures". *ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6, No. 3*, pages 138–151, May 1998.

[2] Bagnall P., Briscoe R., Poppitt A. "Taxonomy of Communication Requirements for Large-scale Multicast Applications". *Internet Draft, draft-ietf-lsma-requirements-02.txt*, Nov. 1998. Work in progress.

[3] Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W. "An Architecture for Differentiated Services". *Internet Draft, draft-ietf-diffserv-arch-02.txt*, Oct. 1998. Work in progress.

[4] Braden R., Clark D., Shenker S. "Integrated Services in the Internet Architecture: an Overview". *RFC 1633*, Jun. 1994. Work in progress.

[5] Bradner S. "Internet Protocol Multicast Problem Statement". *Internet Draft, draft-bradner-multicast-problem-00.txt*, Sep. 1997. Work in progress.

[6] Bradner S. "Internet Protocol Quality of Service Problem Statement". *Internet Draft, draft-bradner-qos-problem-00.txt*, Sep. 1997. Work in progress.

[7] Fall K., Varadhan K. "ns Notes and Documentation". The VINT Project, Jul. 1999. `http://www-mash.CS.Berkeley.EDU/ns/`.

[8] Faloutsos M., Banerjea A., Pankaj R. "QoSMIC: Quality of Service sensitive Multicast Internet protoCol". *Proc. SIGCOMM 1998*, Sep. 1998.

[9] Maggiorini D., Pagani E., Rossi G.P. "A Testbed Environment for the Performance Evaluation of Modular Network Architectures". *To appear in Proc. 1st IEEE European Conference on Universal Multiservice Networks (ECUMN 2000)*, Oct. 2000.

[10] Pagani E., Rossi G.P. "A Functional Architecture for End-to-end Quality-of-Service in a Multi-domain Network". *To appear in Proc. 1st IEEE European Conference on Universal Multiservice Networks (ECUMN 2000)*, Oct. 2000.

[11] Schulzrinne H., Casner S., Frederick R., Jacobson V. "RTP: A Transport Protocol for Real-Time Applications". *RFC 1889*, Jan. 1996. Work in progress.