Computer Networks 54 (2010) 1389–1403

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A new design for end-to-end proportional loss differentiation in IP networks

Pablo J. Argibay-Losada *, Andrés Suárez-González, Cándido López-García, Manuel Fernández-Veiga

Departamento de Ingeniería Telemática, Campus Universitario s/n, Vigo, Spain

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form 20 November 2009
Accepted 2 December 2009
Available online 11 December 2009
Responsible Editor: A. Pitsillides

Keywords:
Quality of service
Loss probability
End-to-end performance
Network architecture
Proportional differentiation

ABSTRACT

This paper describes the algorithms and the architecture of a network able to provide end-to-end proportional packet loss probabilities at the flow level. We show that the combination of a simple classification technique at the sources, and a network core having two internal service classes, is sufficient to achieve proportional service without the need to deploy coordinated, complex per-hop scheduling schemes or signaling protocols, which is the conventional approach. The proposed architecture is complementary to any differentiation algorithm used by the routers. Our results show that any network endowed with some internal service classes with respect to packet loss probabilities can be exploited to build a set of external service classes with end-to-end and per-flow guarantees.

1. Introduction

Packet forwarding in the Internet substantiates the best-effort service model, whereby routers do not keep state information for any of the active traffic flows and every packet receives the same common service. This key architectural principle is behind the unparalleled growth in size, bandwidth and data types carried by the network, but precludes better resource allocation for applications with quality of service (QoS) demands.

The fundamental problem for providing multiple service classes in a packet network has always been the scalability of the architecture and of the routers' algorithms. In the 1990s, IntServ and DiffServ [1] emerged from within the IETF as the two frameworks for building a network core with differentiated services. In IntServ, the applications could obtain even the strictest QoS requirements, since the architecture dictates per-flow, end-to-end resource reservations. As a consequence of this postulate, the routers must maintain and process data and control state for every flow of packets, participate in complex signaling procedures, and cooperate with routers in the same or in other domains in order to support end-to-end service guarantees. Therefore, IntServ cannot be deployed over the global, decentralized Internet, and was never seriously considered for adoption. DiffServ arose as a simpler, more scalable, manageable, and easily deployable solution for service differentiation in IP networks. Its premise is that individual flows with similar QoS requirements can be aggregated in larger traffic groups, called macroflows, that use a certain set of forwarding rules at the core routers, the per-hop-behavior (PHB). Thus, DiffServ can exploit the statistical multiplexing gains of aggregation, it does not require any signaling protocol (i.e., there are no reservations), and it is a distributed architecture, because the PHBs have strictly local semantics. The basic drawback is that the network cannot enforce a specific level of QoS to individual flows within an aggregate. Even the actual quality of service received by a macroflow cannot be predicted accurately.

* Corresponding author. Tel.: +34 986813460; fax: +34 986 812116.
E-mail address: pargibay@det.uvigo.es (P.J. Argibay-Losada).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.12.002

While IntServ and Diffserv represent, respectively, two opposite models of service differentiation, namely absolute QoS performance versus relative differentiation, the paradigm of proportional QoS proposed in [2,3] establishes a trade-off between them. It advocates a per-hop quantitative differentiation where the desired performance of each traffic class is specified as a multiple of that achieved by the adjacent class, in the metric of interest (delay, loss, throughput, or a suitable combination of these). Following these seminal works, many other papers explored independently the possibilities of tuning the scheduling algorithms for delay differentiation, or adapting the buffer management algorithms for loss differentiation (see the discussion and references in [4], for example). Some more recent works investigated designs based on the joint use of both kinds of algorithms for achieving a combination of loss/delay guarantees [5,6], and even to provide support for a limited form of absolute QoS [4,7,8].

A common feature of these proposals is that they leverage suitable per-hop packet forwarding or dropping rules to attain differentiation between a few predefined classes. The challenge of this approach remains in how to extend the model so as to achieve genuine *end-to-end* and *per-flow* differentiation with minimal use of state information at the routers [9], especially if the increasing heterogeneity of networks is to be considered. Performance metrics such as delay or reliability are intrinsically additive along a network path, e.g., the path delay is the sum of the delays in each link (similarly, path reliability is additive after computing logarithms). For these cases, a possible approach to provide end-to-end differentiation over a per-hop QoS infrastructure would be to decompose the end-to-end QoS requirement into a series of local QoS objectives. Unfortunately, the optimal partition of QoS constraints is an intractable (NP-complete) problem [10], and one has to resort to approximate algorithms [11]. Moreover, tunable performance on a per-hop basis does not imply easily controllable end-to-end behavior [9], as with packet losses. For QoS parameters that are not additive, such as bandwidth, decomposition techniques are meaningless. CSFQ [12] presents an architecture supporting approximate end-to-end max-min fair rate allocations to individual flows in a scalable way. Edge routers collect feedback from the network and use this information to check the service rate received by a traffic flow. However, to ensure approximate end-to-end fair bandwidth allocation, CSFQ discards some packets early, thus limiting the utilization, and tags the rest with state variables, so that core routers can extract the context information needed to apply the forwarding rules. The same idea of encoding state information into the packet headers is used in [13] to devise a technique for providing service classes with proportionality in packet loss. Hence, in both cases, packet headers are used for a particular hop-by-hop signaling protocol, which in addition is bound to the existence of routers in the network core operating a coordinated scheduling algorithm. Other works [14] only address the provision of coarse levels of packet loss probability in isolated nodes.

In this paper, we take a different approach from the papers that elaborated on the ideas of [3], which sought the optimal tuning of the scheduling or dropping algorithms as a means to build differentiated services. Indeed, we propose a new approach to obtain in a simple way *end-to-end* and *per-flow* proportional packet loss probabilities. The notion of flow is, in this framework, entirely under the user's control. A flow is defined as a stream of packets with common values in any subset of their IP and TCP/UDP headers, with different flows distinguished by their unique labels (e.g., MPLS, IPv6 or Metro Ethernet). A mapping function associates each label with its service class. Our architecture for end-to-end service differentiation is based on a single key assumption: that the forwarding infrastructure of the network is able to offer two sharply distinct service levels in terms of packet loss, which will be called *premium* and *best-effort*. Thus, the probability of dropping a premium packet is several orders of magnitude lower than a best-effort packet loss. We argue that this condition is easily satisfied by many current scheduling algorithms for a broad class of stochastic traffic models. Then, as long as this separation between the two forwarding classes remains valid, we show that a well chosen probabilistic mapping of the packets in a flow (and in a given traffic class) into the premium and best-effort types is sufficient to achieve the desired proportional end-to-end per-flow loss probabilities. Further, the system architecture is stateless, either in the access, edge or core zones, and the backbone routers do not have to use globally coordinated schedulers. Note that our assumption does not imply the existence of a quantifiable per-hop level of QoS performance. Instead, we build upon pure relative differentiation in the forwarding plane toward individual end-to-end proportional losses. Thus, because there is no need of any protocol to monitor the network state, nor any feedback to keep track of current performance, the architecture is simple, manageable, and distributed. Also, note in passing an interesting property that is frequently overseen: the achievement of packet loss proportionality between a given flow, say f, relative to the best-effort flows leads to a direct bound on the absolute value of the packet loss probability for f. What is more, an absolute QoS guarantee could be offered to flow f if the packet loss probability of the best-effort flows is known, simply by picking the right proportionality ratio between them.

The choice of packet loss as the end-to-end performance metric is justified by several reasons. First, the steady growth in network bandwidth has lowered the contribution of the queueing delays to the end-to-end delay, whereas the packet losses kept unaffected, thus gaining relevance as a performance metric [15]. Second, packet losses are meaningful for applications with real-time constraints [16]. On the other hand, it is well known that packet losses (along with the RTT) determine the throughput performance and bandwidth allocation among a set of TCP-Reno flows [17], so controlling losses could lead to an effective, though indirect, way to provide rate proportionality on a per-flow basis. We shall address this question in a sequel paper. Finally, given the relationship between loss probability and equilibrium rate, the solution presented in this work could be incorporated into the general setting of network utility maximization [18], using losses as link prices.