Flow Aggregation using Dynamic Packet State

Addisu Eshete and Yuming Jiang

Center for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology
addisu.eshete@q2s.ntnu.no ymjiang@ieee.org

1 Introduction

While the Internet is tremendously scalable, it is based on simplistic design that provides only egalitarian best effort service for all traffic types. Two architectural extensions have been proposed, namely Intserv and Diffserv, to enhance the service architecture. While Integrated Services (Intserv) can provide hard Quality of Service (QoS) requirements (delay and bandwidth) to individual flows end to end, its per-flow mechanisms and signaling overhead make it too complex to be readily deployed in the Internet core. On the other hand, Differentiated Services (Diffserv) restores the scalability of the Internet by dropping per flow mechanisms and merging of flows into aggregates. The price paid is the lack of flexible and powerful services at flow granularity levels.

In this work, we plan to address the question: **can we *simultaneously* provide efficient and scalable QoS and also fulfil the requirements of individual constituent flows?**

2 The Techniques– Flow Aggregation and DPS

To provide scalability, we can employ flow aggregation which provides a host of benefits: (1) the number of flows in the core of networks are reduced, and so are the complexity associated with per flow management and operations at core routers, (2) scheduler efficiency of routers can be improved [2], and (3) when the reserved rate of a flow is coupled with delay as in guaranteed rate schedulers [3], flow aggregation can result in tighter bounds of the queueing delay [1, 2]. However, there are known issues with flow aggregation. We outline two known problems: (1) FIFO aggregation of EF traffic in large arbitrary networks may explode the delay bound after a certain utilization threshold [1, 4] and it is not possible to provide delay bounds for high utilization levels in these networks, and (2) flow aggregation usually needs to be nonwork-conserving to be fair to individual constituent flows. Continuous proliferation of very high capacity links means that the first issue may not be the major problem. The main challenge is the impact of (work-conserving) aggregation on the resulting QoS provision of the constituent flows of the aggregate.

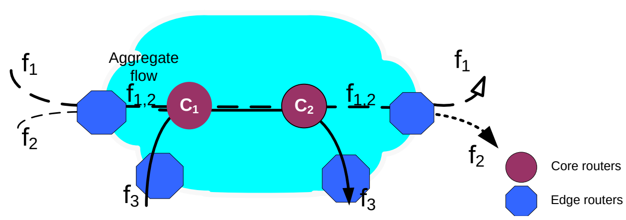
To provide efficient flow aggregation that also satisfies flow requirements, we plan to use *dynamic packet state* (DPS) [5]. Using DPS, per flow states or requirements are inserted into the packet headers at the edge as labels, freeing core routers from the task of maintaining flow states. The states to be encoded are dynamic since the current flow state is recomputed at each node and encoded as the packet traverses through the network. These labels are then used to coordinate the actions of distributed algorithms in the network domain. The labels are ultimately removed at the egress nodes for interoperability to existing architectures.

3 Our Approach

It is important to note we use flow aggregation for scalability, and DPS to make it efficient (work-conserving) and able to fulfill per flow requirements.

A schematic of our conceptual system model is shown in Fig. 1. Flows with similar characteristics are aggregated at the ingress node. We assume that routers have aggregation capability at their output ports. Core routers C_1 and C_2 are aware of the aggregate flow $f_{1,2}$ but they have no knowledge of the constituent flows f_1 and f_2. The challenge is to preserve the quality of service requirements of constituent flows in the network core without the core routers knowing the composition of the aggregate flow.

Fig. 1. Flow aggregation. f_1 and f_2 are constituents of flow $f_{1,2}$ while f_3 is cross traffic.



We propose a *virtual GR* which is a variant of GR (Guaranteed Rate) scheduling [3] modified for aggregation. At entrance to the domain, ingress node ($j = 1$), we start by encoding the $vGRC$ for packet i of the flow as follows:

$$e_1^i = max\left\{a_1^i, vGRC_1^{i-1}\right\} \quad \text{where } vGRC_1^i = e_1^i + l^i/r \tag{1}$$

At inner node j, when packet p^i arrives it is assigned:

$$\begin{aligned} e_j^i &= vGRC_{j-1}^i \\ vGRC_j^i &= e_j^i + l^i/r \end{aligned} \tag{2}$$

where r is flow rate at simple scheduler and sum of flow rates at an aggregator. The QoS provided to the individual flows using this scheme should be comparable to the per flow schemes without aggregation. We believe the result of this work will be very useful to provide QoS guarantees to flows in a scalable manner in the future Internet.

References

1. A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In *Proceedings of Quality of Future Internet Services*, pages 1–13, 2000.
2. J. A. Cobb. Preserving quality of service guarantees in spite of flow aggregation. *IEEE/ACM Trans. on Networking*, 10(1):43 – 53, Feb 2002.
3. P. Goyal, S. S. Lam, and H. M. Determining end-to-end delay bounds in heterogeneous networks. *Multimedia System*, 5(3):157–163, May 1997.
4. Y. Jiang. Delay bounds for a network of guaranteed rate servers with fifo aggregation. *Computer Networks*, 40(6):683–694, 2002.
5. I. Stoica. *Stateless Core: A Scalable Approach for Quality of Service in the Internet*. PhD thesis, CMU, Pittsburgh,USA, Dec 2000.