A Proposal for an Asymmetric Best-Effort Service

Paul Hurley <paul.hurley@epfl.ch>, Jean-Yves Le Boudec <leboudec@di.epfl.ch>
ICA, Ecole Polytechnique Fédérale de Lausanne (EPFL).

Abstract

We propose Asymmetric Best-Effort, a novel service to provide a "throughput versus delay jitter" differentiated service for IP packets. With this service, every best effort packet is marked as either *Green* or *Blue*. *Green* packets, typically sent by real-time applications such as interactive audio, receive more losses during bouts of congestion than *Blue* ones. In return, they receive less delay jitter.

Both *Green* and *Blue* services are best-effort. The incentive to choose one or other is based on the nature of one's traffic and on traffic conditions. If applications are TCP-friendly, those sending *Blue* packets will receive more throughput but also more delay jitter, than they would if they sent *Green* packets for a given network state and path.

1 Introduction

We propose a new Asymmetric Best-Effort service. It partitions IP packets into *Green* packets, which are given low delay jitter guarantees through the network without reservation, and *Blue* packets which receive fewer losses during bouts of congestion than *Green* ones.

Provided sources are "TCP-friendly", those who choose to be *Blue* receive higher throughput than they would if they had chosen to be *Green*. It is important to emphasise that these are both best-effort services, and the incentive to choose one or other is based on the nature of one's traffic with overall benefit for both traffic types.

Each packet is either *Green* or *Blue*. *Green* packets would usually be interactive traffic such as Internet Telephony where packet transfer from end to end must be short and delay jitter significant. *Blue* packets are typically non-interactive traffic such as TCP traffic whose end to end delay can be variable and the goal is minimisation of overall transfer time. We leave open to subsequent definition how the *Green* and *Blue* distinction should be made.

The amount of negative feedback (e.g. packet losses) received by *Green* traffic is greater than that received by *Blue* traffic. The admitted *Green* packets are given a shorter queueing delay. The network-level quality of service, packet loss and delay jitter, received by one of the traffic types cannot be classified as being better than the other.

No rate reservation is assumed. During a silent period of a given traffic type, the other can make use of the whole bandwidth. Traffic management and charging practices remain essentially the same as for a single class best-effort network. Unsatisfactory trade-offs between the different buffer size requirements of real and non-real time traffic can also be avoided.

We assume that *Green* and *Blue* sources are "TCP-friendly" [1], i.e. they do not send more than a TCP source would for the same conditions of loss. Although the approach described remains valid when using some form of explicit negative feedback such as Explicit Congestion Notification (ECN), we consider here only the case where packet loss implicitly provides feedback.

2 Router Implementation

Asymmetric Best-Effort traffic control can be supported within an IP router by Packet Admission Control (PAC) and differential scheduling. The PAC manages the queue by dropping packets whenever necessary or appropriate, acceptance being biased in favour of *Blue* packets. The scheduler determines which packet, if any, from the buffer should be sent next, with bias in favour of *Green* packets.

We now describe one router implementation which was inserted and simulated in ns [5]. The service goal is the distribution of throughput such that a *Blue* flow would receive approximately β times as much throughput as a *Green* one that shares the same path.

The scheduling is Earliest Deadline First (EDF) [4]. Each packet is assigned a finishing service time deadline, a tag, and the packet currently having the lowest value is served first (i.e. earliest deadline).

Each *Green* packet arriving is assigned a finishing service time deadline equal to the arrival time t. A *Blue* packet is assigned a time equal to the arrival time plus a constant D, namely $t + D$. Using EDF, rather than a plain priority scheme in which *Green* packets would always be served first, has the advantage of preventing service starvation for *Blue* traffic.

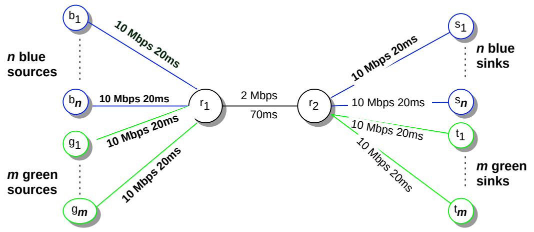

Figure 1: Simulation Network used in Illustration of Service

The PAC comprises a modified version of Random Early Detection (RED) [3] in which the dropping probability for *Blue* packets is the usual RED dropping probability p while for *Green* packets it is αp.

In addition, to ensure *Green* packets are given a sufficiently small delay they must pass a second acceptance decision, called *Green* delay control. For this implementation, a *Green* packet is accepted into the queue only if the number of packets, N, in the queue with tag of less than the current time, is less than or equal to a given system permitted number N_g. The PAC algorithm can now be summarised as follows:

```
For each packet arrival to output port:
  if buffer full
    drop packet
  else if Blue
    drop with random probability p
    if not dropped
      deadline = now + D
      accept packet
  else Green
    drop with random probability alpha*p
    if not dropped
      if N > Ng
        drop packet
      else
        deadline = now
        accept packet
```

α is varied such that it drives the effective measured ratio of *Green* to *Blue* drop ratios α_m towards the target ratio of *Green* to *Blue* loss ratio α_t. For this iteration, the mechanism to drive α_m towards α_t is simple and given by $\alpha = \alpha_t^2/\alpha_m$. α_t is chosen to be of the order of β^2, as described in [2], in order to attain the service goal. As ongoing work, the target ratio α is currently being adjusted to allow for the fact that the lower queueing delay causes *Green* flows to have a lower round-trip time and thus achieve a higher throughput than one expects if the round-trip time was the same.

3 Service Illustration

We show that an application that requires low delay jitter and thus sends *Green* packets does receive it but at the expense of lower throughput. Conversely, an application that does not care about jitter receives overall reduced end to end transfer delay.

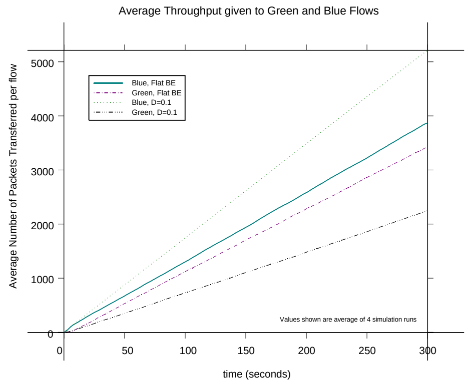

Figure 2: Average of the average number of packets received per *Green* and *Blue* flow with/without Asymmetric Best-Effort ($n = m = 10$).

The network used in this simulation is shown in Figure 1. n *Blue* and m *Green* flows share a bottleneck and the same nonqueueing delay. Router r_1 facilitates the service by the implementation as described in Section 2. The *Blue* sources are TCP Reno.

Since a *Green* source would typically be rate rather than window based, and not necessarily concerned with loss recovery given its real-time nature, it uses a simple transport protocol designed to approximate TCP friendliness as described in [2].

For all simulation results shown, $D = 0.1$, $\beta = 3$ and $N_g = 10$. Packet sizes were a constant 1000 bytes, and the buffer size at r_1 was 60 packets. The RED parameters chosen were $min_{th} = 0$, $max_{th} = 40$, $max_p = 0.2$ and $w = 0.02$ which are not the usual range of parameters used. With more typical RED parameters there are many forced losses and hence less control over the dropping differential.

The first simulation results we show are for $n = 10$ and $m = 10$. Figure 2 shows the average over 4 simulation runs of the average number of packets transferred by each traffic type at a given time t for the cases with and without Asymmetric Best-Effort. The throughput given to *Blue* traffic is clearly higher in the asymmetric case. The worst-case interval for 95% confidence was 1.30 packets. Figure 3 shows the better distribution of queueing delays experienced by *Green* traffic at router r_1 when using Asym Best-Effort.

Overall benefit for each source is achieved. A source that sees value in low delay jitter will choose to be *Green*. This enables file transfer oriented applications, which choose to be *Blue*, to achieve higher throughput.

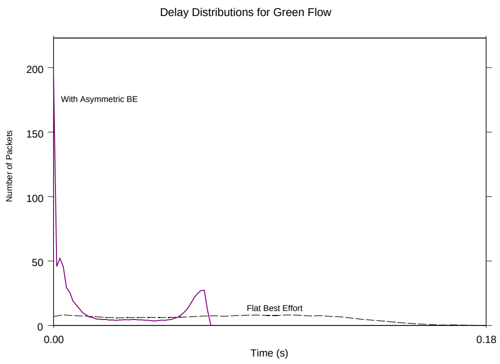

Figure 3: Queueing Delay for *Green* packets with/without Asymmetric Best-Effort. ($n = m = 10$).

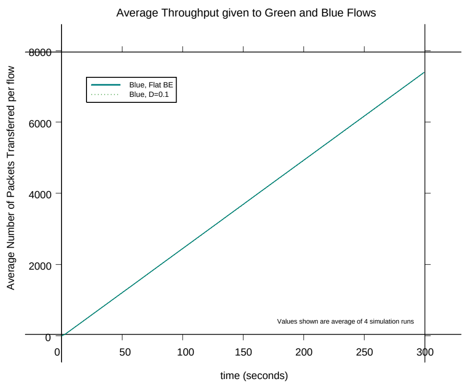

Figure 4: With all *Blue* traffic, Asymmetric Best-Effort resorts to the flat case.

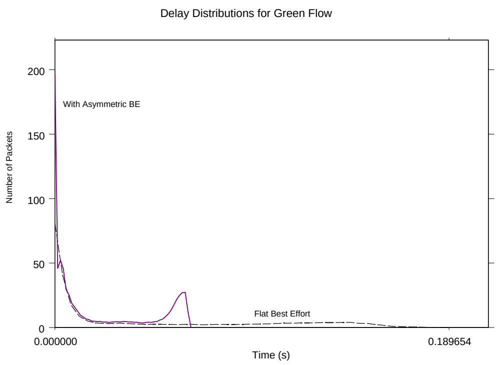

Figure 5: If all *Green* traffic, a lower delay with Asymmetric Best-Effort is still received.

Figure 4 shows, in the case of all-*Blue* traffic ($n = 10, m = 0$), that the service reverts to regular flat best-effort. The worst-case interval for 95% confidence here was 0.3561 packets.

Figure 5 shows, in the case of all-*Green* traffic ($n = 0, m = 10$), the queueing delay distribution for *Green* traffic for the service is still better than in flat best-effort, due to the effect of the green dropping control, which permits only a small number of green packets in the queue.

4 Conclusions

We have described a simple but powerful service which enables best-effort traffic to receive requirements closer to its traffic desires yet to everyone's overall benefit. It decouples delay jitter objectives from loss objectives with no concept of reservation or signalling and no change to traffic management or charging. Dimensioning the network is also potentially simpler since one would no longer need to choose a buffering compromise to suit both types of traffic.

It should be stressed that Asymmetric Best-Effort is a new service in its own right and not a substitute for reservation or priority services.

The service choice of *Green* or *Blue* is self-policing since the user/application will be coaxed into choosing one or the other or indeed a mixture of both, based on its traffic profile objectives.

We provided a first implementation of the service, of which further optimisation of the control loop and the setting of RED parameters remains. In particular, the dropping probability for *Green* packets needs to be increased to compensate for the shorter queueing delay resulting in higher throughput.

References

[1] TCP friendly web site. http://www.psc.edu/networking/tcp_friendly.html

[2] Paul Hurley, Jean-Yves Le Boudec, Maher Hamdi, Ljubica Blazevic, Patrick Thiran. The Asymmetric Best-Effort Service. *SSC Technical Report SSC/1999/003, EPFL*, February 1999.

[3] Floyd, S., and Jacobson, V. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, V.1 N.4, August 1993, p.397-413.

[4] R. Guerin and V. Peris. Quality-of-Service in Packet Networks - Basic Mechanisms and Directions. *Computer Networks and ISDN Systems. Special issue on multimedia communications over packet based networks*, 1998.

[5] ns v2 simulator. See http://www-mash.cs.berkeley.edu/ns/