TCP adaptation to EDS

Benjamin Gaidioz, Pascale Primet

To cite this version:

Benjamin Gaidioz, Pascale Primet. TCP adaptation to EDS. [Research Report] RR-5121, INRIA. 2004. inria-00071462

HAL Id: inria-00071462
https://hal.inria.fr/inria-00071462

Submitted on 23 May 2006

TCP *adaptation to* EDS

Benjamin Gaidioz, Pascale Primet

N° 5121

February 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5121--FR+ENG

TCP adaptation to EDS

Benjamin Gaidioz[*], Pascale Primet[†]

Thème 1 — Réseaux et systèmes
Projet RESO

Rapport de recherche n° 5121 — February 2004 — 11 pages

Abstract: The TCP/IP stack has been mainly designed for elastic traffic (file transfers). It is nowadays recognized that it is not able to efficiently support traffic patterns with completely differing requirements (e.g. applications with delay requirements). Service differentiation at the flow agregate level (DiffServ) is a promising way to implement some form of IP QoS because it is robust and scalable. The EDS PHB is a Diffserv PHB based on both loss rate and delay proportional differentiation. In this article, we start from a network layer implementing the EDS service differentiation and present an marking strategy for TCP. This strategy leads to a diminution of the number of timeouts and the standard deviation of the individual throughput of multiple connections.

Key-words: IP quality of service, service differentiation, Equivalent Differentiated Services, TCP, fairness

---

[*] Université Lyon 1, LIP RESO (UMR CNRS-INRIA-ENS-UCBL N°5668)
[†] LIP RESO (UMR CNRS-INRIA-ENS-UCBL N°5668), INRIA

Adaptation de TCP à EDS

Résumé : La pile de protocoles TCP/IP a été conçue principalement pour transporter du trafic « élastique » (transferts de fichier). Aujourd'hui, il est admis qu'elle ne convient pas pour de nouvelles catégories de trafic aux besoins très différents (par exemple des flux temps-réel avec des contraintes de délais). La différenciation de services au niveau d'agrégats de flux (DiffServ) est une architecture prometteuse pour mettre en oeuvre une forme de qualité de service IP car elle est robuste et s'étend bien à un grand nombre de noeuds. EDS est un PHB Diffserv qui s'appuie sur la différenciation proportionnelle en délai et taux de perte. Dans cet article, en nous appuyant sur le modèle de différenciation de service EDS, nous présentons une stratégie adaptative pour TCP. Cette stratégie permet de diminuer le nombre de timeouts des connexions et de diminuer l'écart-type des débits.

Mots-clés : qualité de service sur IP, différenciation de services, Equivalent Differentiated Services, TCP, équité

1 Introduction

The network layer of Internet (IP) provides a simple, robust and effective packet forwarding service. The TCP/IP stack has been mainly designed for elastic traffic (e.g. FTP). It is nowadays recognized that it is not able to efficiently support new traffic patterns with different requirements in terms of speed, latency and reliability (e.g. real-time applications, interactive applications, bulk transfers, etc.). It is commonly accepted that IP needs to be extended with a form of quality of service (QoS). In the case of IP, adding QoS has to be done carefully in order to maintain the efficiency and robustness of the network.

The DiffServ architecture [2] is convincing as a QoS approach on a large scaled network. Its principles keep the network layer simple and robust. In the core network, IP packets are expected to be marked with a specific *class identifier*. This identifier selects a forwarding treatment met by the packet each time it crosses a router. From the core network point of view, there is a very small number of classes, there is no resource reservation between hops. Marking is done at the edge routers, where the number of flows is sufficiently low to apply marking rules without losing much performance.

A characteristic that makes TCP/IP so appealing is its ease of use: a host can plug into the network and use it immediately. According to its design philosophy [4], IP attempted to provide a basic *building block* out of which a variety of types of service could be built. The decision was an extremely successful one, which allowed the Internet to meet its most important goals.

In this report, we rely on the implementation of the EDS service differentiation [10, 9] at the IP level (see in the technical report [9] for details). This architecture provides a best-effort service differentiation which is not sufficient to satisfy most of the applications needs. Thus, it needs to be extended with end-to-end protocols which use the services it provides in order to ensure some higher level guarantees to applications.

We start the article with a quick reminder on EDS (sect. 2). Then we present a marking strategy for TCP in sect. 3 as well as simulations. In sect. 5, we present related work. Finally, in sect. 6, we give conclusions and future work.

2 Quick overview of EDS

Fig. 1 gives the intuition of the concept of Equivalent Differentiated Services. It shows both the service provided by a best-effort router and an EDS router. Roughly, the EDS router provides a set of N classes, each one experiencing a different level of performance in both queuing delay and loss rate. Considering a single performance criteria (delay or loss rate), the range of performance is centered around an average performance, which is the performance all packets would have obtained through a plain best-effort router. Thus, the performance is directly linked to the router load. Moreover, there is an asymmetry between delay and loss rate performance: The class which obtains the i^{th} best performance in delay obtains the i^{th} worst performance in loss rate.

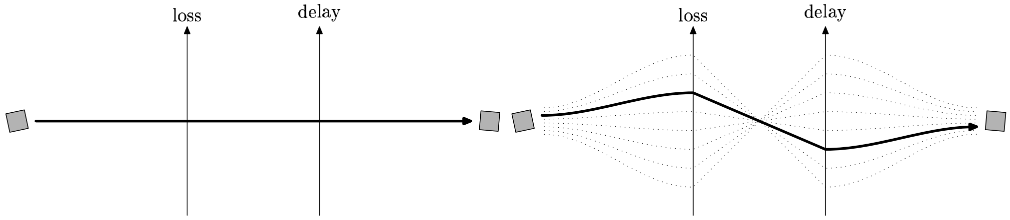

Figure 1: A packet (gray square) is crossing a best-effort router (left) or an EDS router (right). Through the best-effort router, the packet experiences a given queuing delay and a given loss probability. Through an EDS router, the packet is member of a service class among N classes, where it experiences a specific queuing delay and a specific loss probability, both being relatively better or worst than the performance through a best-effort router.

The EDS service differentiation proposal has been designed by starting from the Internet protocol design principles which gave to IP its robustness, ease of deployment and ease of use. Thus, EDS provides *best-effort* service differentiation.

Stronger guarantees on the plain best-effort IP are implemented in end-to-end protocols. For example, TCP guarantees reliability. These protocols use the building block provided by IP to provide a specific service matching the need of specific applications. Since we have defined a best effort service differentiation system from the same design rules, the natural way to implement stronger QoS guarantees has to be done in the protocol layer.

3 Marking strategy for TCP

TCP has a well known algorithm that regulates its congestion window in an AIMD manner [11, 1] which provides some interesting properties in terms of efficiency and fairness between several connections [3]. The marking strategy we propose consists in improving a bit the efficiency and fairness by:

- reducing the number of timeouts during a transfer,
- protecting the connection by large load,
- reducing the standard deviation of the rate of a group of connections.

We took the standard deviation of the rate of connections as an indicator for fairness. If it is large, then it means that there is a large variation in terms of performance between connections. The narrower the standard deviation, the best the fairness.

The marking algorithm is shown as algorithm 1. The larger the number of RTT a connection manage to run without detecting a loss, the more lucky the connection is. It manages to increase its window size more than a connection that experiences some losses. When a connection experiences a loss, it needs to be temporary protected because in the case the retransmitted packet is lost or it experiences a second loss, the connection will really suffer. The rules are the following.

3.1 Marking rules

1. a connection starts in class 1, the one with the lowest loss rate,
2. after having run successfully (without losses) during M RTT, a connection moves from its class i to class $i+1$. The connection may continue to increases its class identifier until it reaches the highest class.
3. when a connection experiences a loss and detects it, it moves back to class 1.

The TCP AIMD algorithm is not modified. This just consists in adding some specific marking rules in TCP.

```
$class \leftarrow 1$
$date\_last\_drop \leftarrow now$
while there is data to send do
  send packets
  if drop is detected then
    $date\_last\_drop \leftarrow now$
    $class \leftarrow 1$
  end if
  $class \leftarrow class + (date\_last\_drop - now) \div (M \times \text{RTT})$
  if $class > N$ then
    $class \leftarrow N$
  end if
end while
```

Algorithm 1: Marking strategy for TCP.

3.2 Handling of reordering

This marking strategy that consists in moving from a class to an other may lead to packet reodering. Packet reordering can lead to fake loss detection and thus spurious retransmissions. In our implementation, we have modified the fast retransmit optimization so that it does not react when selective acknowledgements (usually indicating a loss) acknowlege quicker packets. This is done by checking the class a packet was sent before triggering its retransmission. If duplicated ACK occured because of acknowledgment of quicker packets, then the fast retransmission is not triggered.

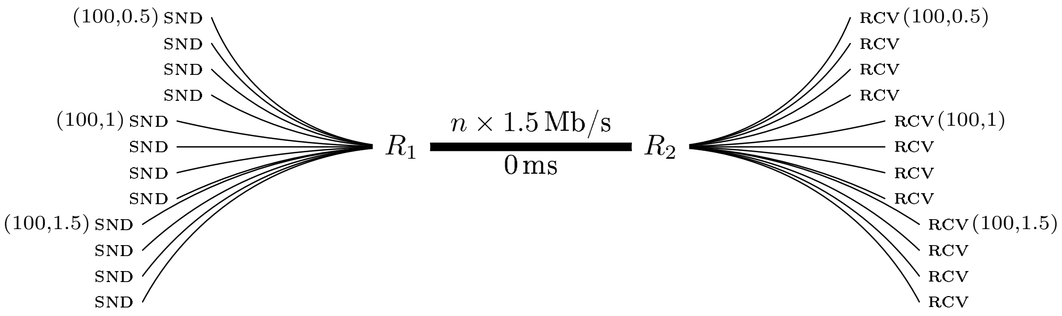

Figure 2: Experimental network.

4 Experimental results

The algorithm has been tested in simulation. In the experiment, we suppose that N connections share a link, while no random traffic crosses it (the link becomes congested because of the AIMD algorithm). There is a single router which implement either best-effort (RED) or EDS.

4.1 Experimental network

The experiments are run over the network shown on fig. 2. There is a single router that is crossed by traffic generated by TCP connections. The bandwidth of the shared link depends is proportional to the number n of connections. This permits to avoid running the experience in completely under/over-provisioned cases. The link itself has no propagation delay.

There are three levels of connections with each time a different link latency. The link latency is either $0.5 \times L$, or $1 \times L$, or $1.5 \times L$ with L being a varying parameter of the experiment. The EDS configuration is the following: There are 16 classes. The spacing between classes because of differenciation is constant. There is a ratio of 8 (not 16) between the first and the last class.

In this set of experiment, we vary the number n of connections and the latency parameter L. This leads to a large number of results one cannot show easily. We have chosen to represent the results linearly, with no hierarchy between the experiments. We order the different configurations (n, L) so that there is an increase of the "performance" (rate, timeouts, etc.) from the RED point of view. On the same graph, we show then both the performance obtained with RED and the performance obtained with EDS.

4.2 Number of timeouts

Fig. 3 shows the average number of timeouts experienced by TCP connections. It is obvious that the number of timeouts is lower with EDS than with RED. This is an expected improvement obtained by the marking strategy. It is important to note that the number of timeouts

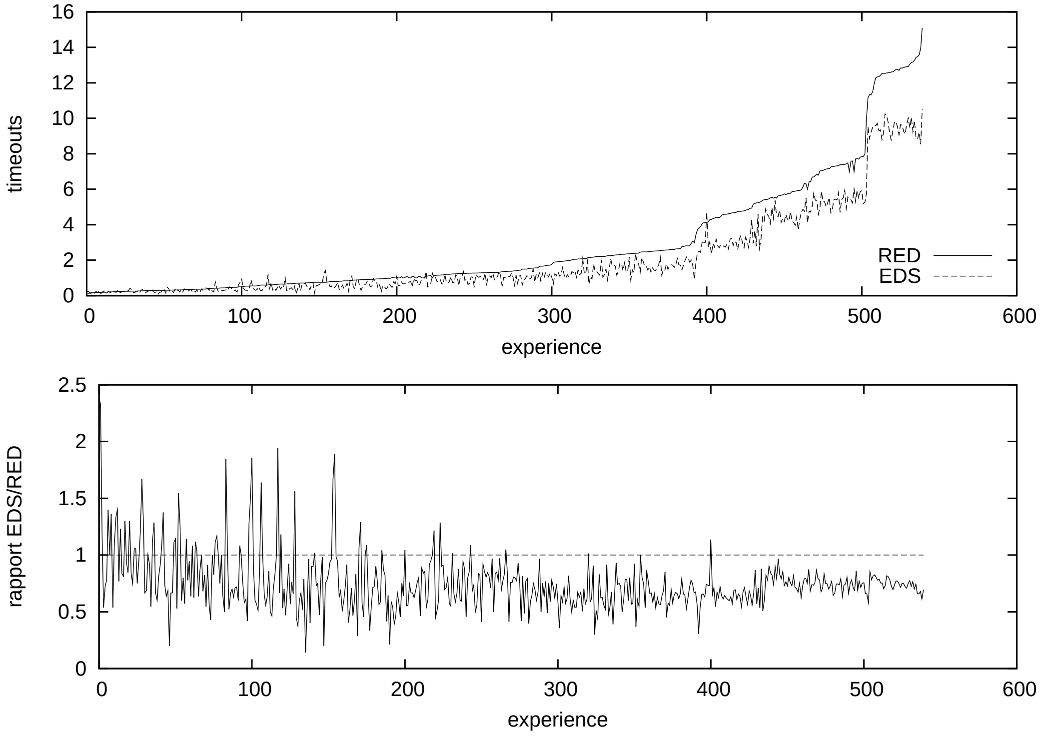

Figure 3: Top: quantity of timeouts. Bottom: ratio EDS/RED.

with RED or EDS is relatively low because we use a very efficient version of TCP (considering loss recovery) which is the SACK-TCP version.

4.3 Goodput

Fig. 4 shows the average goodput obtained during the experiment. Using both RED or EDS, their is no improvement.

- There is no improvement because in this kind of network, several TCP connections can use efficiently the network and we could not expect an improvement of the use simply with a marking strategy.
- The fact that the standard deviation is lower is due to the reduced number of timeouts (see the next section) and the fact that the strategy protects connections which met issues one RTT sooner.

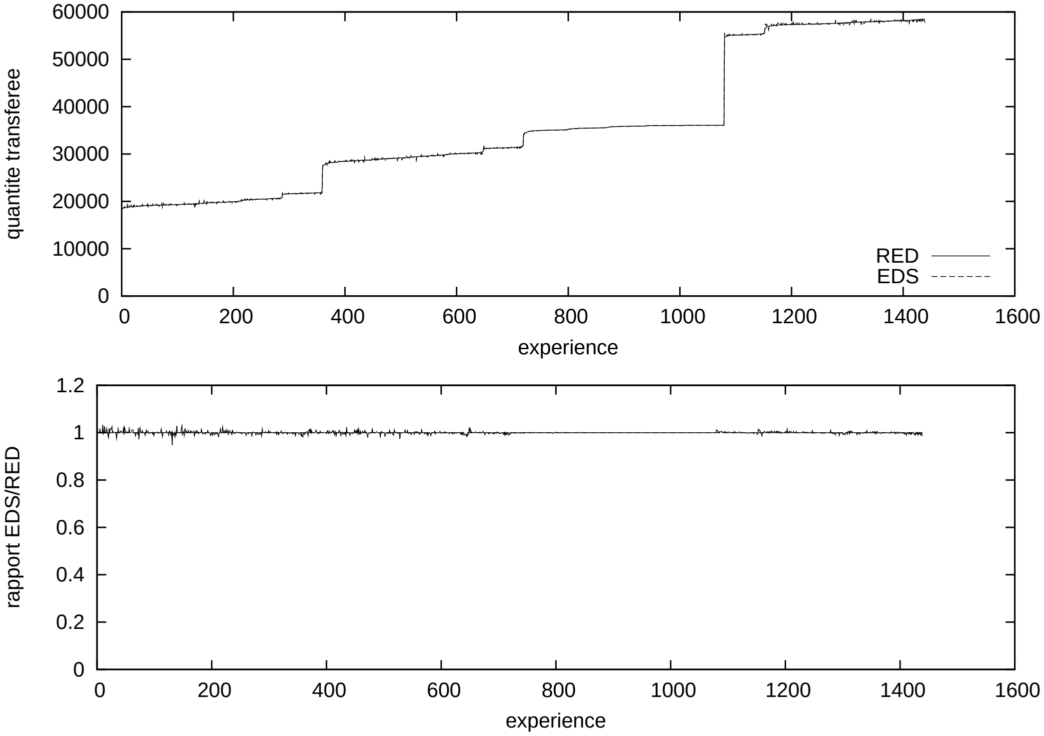

Figure 4: Top: quantity of bytes transfered. Bottom: ratio EDS/RED.

4.4 Standard deviation

Fig. 5 show the standard deviation of the number of bytes transfered by connections. Graphs show that the standard deviation obtained with EDS is lower than with RED.

5 Related Work

Similar work is conducted on top of different service differentiation systems.

- The authors of proportional differentiated services [5] have seen the need to match stronger QoS requirements from the end-to-end viewpoint since proportional differentiation does not provide absolute guarantees. An adaptive protocol [6] has been designed to dynamically select the best class and then provide an absolute guarantee in delay.

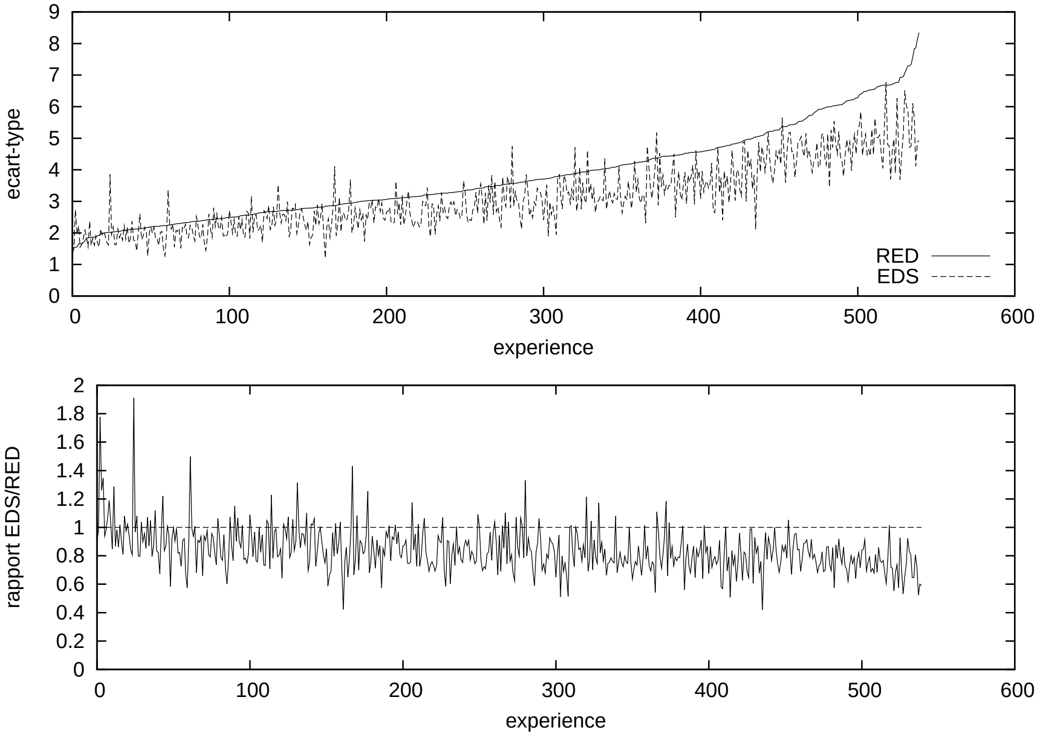

Figure 5: Top: standard deviation. Bottom: ratio EDS/RED.

- The TCP-friendly differentiated services marking [7] is based on the same way to consider service classes as a building block protocols can use with a fine grained adaptation algorithm. The system is based on the standard *assured* service where classes get different drop level probability (no delay differentiation). Their objective is similar to us since they do not aim at providing strong guarantees but improving the overall performance (they prove that they obtain less timeouts).

In both cases, the underlying service differentiation layer provides privileged classes. The architectures must be deployed with an access control system (or marking has to be done by a trustworthy border entity) to ensure a cordial use of services.

6 Conclusion and future work

We have presented both the EDS best-effort service differentiation system and a transport protocol built on top of it.

The EDS system has been designed by mapping the IP design philosophy to service differentiation. There is a need to implement marking strategies in transport level protocols. In this report, we have presented a adaptation of TCP to EDS. The marking strategy included in TCP improves the fairness of the rate sharing between several connections by decreasing the number of timeouts. At the end of the day, the standard deviation is lower through an EDS network, which means that the bandwidth was shared a bit better.

From a more practical point of view, since we implemented EDS with proportional differentiation schedulers in Linux [8], we are also going to start the implementation of the protocol presented in this article and see how it runs in real life.

References

[1] Mark Allman, Vern Paxson, and W. Richard Stevens. TCP congestion control. Internet Request For Comments RFC 2581, Internet Engineering Task Force, April 1999.

[2] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Internet Request For Comments RFC 2475, Internet Engineering Task Force, December 1998.

[3] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and* ISDN *Systems*, 17(1):1–14, June 1989.

[4] David D. Clark. The design philosophy of the DARPA internet protocols. In *Proceedings of Special Interest Group on data Communication (*SIGCOMM*)*, number 4 in Computer Communication Review, pages 106–114, Stanford, CA USA, August 1988. ACM, ACM Press.

[5] Constantinos Dovrolis and Parameswaran Ramanathan. A case for relative differentiated services and the proportional differentiation model. IEEE *Network*, 13(5):26–34, September 1999.

[6] Constantinos Dovrolis and Parameswaran Ramanathan. Dynamic class selection: From relative differentiation to absolute qos. In *Proceedings of the International Conference on Network Protocols (*ICNP*)*. IEEE, November 2001.

[7] Azeem Feroz, Amit Rao, and Shivkumar Kalyanaraman. A TCP-friendly traffic marker for IP differentiated services. In *Proceedings of International Workshop on Quality of Service (IWQoS)*, Pittsburgh, PA USA, June 2000. IEEE/IFIP.

[8] Benjamin Gaidioz, Mathieu Goutelle, and Pascale Primet. Implementation of IP proportional differentiation with waiting-time priority and proportional loss rate dropper in linux. Technical Report RR-4511, INRIA, August 2002.

[9] Benjamin Gaidioz and Pascale Primet. The equivalent differentiated services. Technical Report RR-4387, INRIA, February 2002.

[10] Benjamin Gaidioz and Pascale Primet. EDS: A new scalable service differentiation architecture for internet. In *Proceedings of International Symposium on Computer Communication (*ISCC*)*, pages 777–782, Taormina, Italy, July 2002. IEEE.

[11] Van Jacobson. Congestion avoidance and control. ACM *Computer Communication Review*, 18(4):314–329, 1988.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399