The Equivalent Differentiated Services Model

Benjamin Gaidioz, Pascale Primet

To cite this version:

Benjamin Gaidioz, Pascale Primet. The Equivalent Differentiated Services Model. [Research Report] RR-4387, LIP RR-2002-09, INRI, LIP. 2002. inria-00072201

HAL Id: inria-00072201
https://hal.inria.fr/inria-00072201

Submitted on 23 May 2006

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Equivalent Differentiated Services Model

Benjamin Gaidioz, INRIA RESO, RESAM

Pascale Primet, INRIA RESO, RESAM

No 4387

Février 2002

THÈME 1

Rapport de recherche

ISSN 0249-6399

The Equivalent Differentiated Services Model

Benjamin Gaidioz, INRIA RESO, RESAM
Pascale Primet, INRIA RESO, RESAM

Thème 1 — Réseaux et systèmes
Projet Reso

Rapport de recherche no 4387 — Février 2002 — 10 pages

Abstract: It is now quite agreed that the IP layer should provide more sophisticated services than the simple best-effort service to meet the application's quality of service requirements. Different proposals for improving the IP stack have been proposed but their deployment on experimental network shows different scaling problems. In this paper we present a new differentiated service scheme called "Equivalent Differentiated Services". The main features of the EDS scheme is that it allows to differentiate an arbitrary number of traffic classes without giving an absolute better service to any. EDS is simple, robust and scalable and provides no absolute guarantees in delay or loss rate to individual flows or aggregates. EDS acts as a network layer protocol and needs some adaptation done by an end-to-end transport layer. We present the EDS model, then we examine the implementation, simulation and validation issues.

Key-words: IP QoS, service differentiation, different but equal models, non-elevated models.

Le modèle « Equivalent Differentiated Services »

Résumé : Il est presque universellement admis le service *best-effort* offert par la couche IP ne suffit plus aujourd'hui pour satisfaire les besoins des applications en terme de qualité de service. Les différentes propositions d'architecture pour l'ajout de qualité de service (QoS) à Internet se sont heurtées pour la plupart à des difficultés de déploiement à grande échelle. Dans cet article, nous présentons le modèle « Equivalent Differentiated Services » (services différenciés équivalents). L'intérêt principal de EDS est que la différenciation est pratiquée sur un nombre arbitraire de classes sans donner de service meilleur de manière absolue à l'une d'elles. EDS est d'une conception simple, robuste et peut s'étendre facilement à un grand réseau. Les garanties offertes en délai et taux de perte sont d'ordre relatif (et non absolu) entre les aggrégats de flux d'une même classe. EDS est une sorte de couche réseau qui nécessite un système d'adaptation aux extrémités du réseau. Nous présentons EDS puis sa mise en œuvre, quelques résultats en simulation et enfin les perspectives en terme de validation.

Mots-clés : QoS/IP, différenciation de service, modèles differents mais égaux.

1 Introduction

Internet has become a critical infrastructure for a variety of applications generating traffics that vary in their sensitivity to network performance. It is now agreed that the network layer should provide more sophisticated services than the simple best-effort service to meet the application quality of service requirements. According the IP design philosophy [7], IP attempted to provide a basic *building block* out of which a variety of types of service could be built. The datagram is a building block and not a service in itself. The decision to use the datagram was an extremely successful one, that allowed the Internet to meet its most important goals successfully. But the TCP/IP stack, mainly designed for elastic traffic, is not able to support traffic with completely differing requirements in speed, latency and reliability. Since the *best-effort* service provided by the IP layer is a flat packet forwarding service, TCP [19] and other end-to-end protocols like RTP/RTCP [1] have to implement some high-level functionalities (packet reordering, error control, time-stamping, error control, etc.) to provide the application a service that matches their performances needs. The question today is: which building block can replace the IP forwarding initial mechanism to allow the development of an efficient "multi-service Internet" that matches the requirements of the new applications?

The problem of the quality of service can be summarized as finding a solution to regulate usage of the network during congestion. If there is no congestion, there is presumably no queue of packets, no delay, no packet loss, full capacity and low jitter. Then, one solution to provide enough resources to high end applications is overprovisioning. In this approach, the IP layer remains the same, users only wait operators to provide more bandwidth. But overprovisioning cannot satisfy all the QoS requirements, specially because it is not an end-to-end solution. Congestion still occurs, at the edge nodes. For example, LANs or end-systems network stacks may be misconfigured. Consequently, the quality of service observed by a flow can be very low (long delay, high packet loss rate) or unstable despite the fact it crosses overprovisioned backbones. For us, overprovisioned networks are a partial solution to QoS. Queue management systems such as RED [10] and ECN [21] mechanisms tend to improve the best-effort service by having more control on congestion. However, they do not serve differently the applications and do not take into account the heterogeneity of needs. Indeed, the QoS needs of the different traffics in the Internet are heterogenous. For example, real-time flows are sensitive to delay and delay variation; a file transfer needs a low packet loss rate and a high throughput. The IntServ group identifies two main types of traffic: elastic and real-time [4, page 12], both including several subtypes with different requirements.

In sect. 2 of this paper, we briefly analyze different sophisticated propositions for adding QoS in the Internet and we show their main drawbacks which, probably, slow their large acceptance and deployment. In sect. 3 we present a new model for adding QoS to IP which aims to conform as much as possible the IP philosophy. Our goal is to design a building block which permits service differentiation but remains as simple, robust and scalable as possible. This model, called "Equivalent Differentiated Services" (EDS), can be classified as a "non-elevated service" [23] in the sense that it proposes "different but equal" service classes and requires no policing and little operational change. We argue that such a model needs to rethink (or at least to adapt) the transport layer. Sect. 3.4 describes the scheduling mechanism of EDS. Sect. 4 shows ongoing work on adapting the transport layer to EDS with an adaptative real-time application. In the final part (sect. 5) we discuss the limits of the model proposed and the research perspectives.

2 Existing solutions for Quality of Service in IP networks

Several techniques for the provision of quality of service (QoS) guarantees in IP networks are developed. Since the early 90's, the IETF proposed standards to include QoS in IP.

The first one is the Integrated Services (IntServ) architecture [25, 22]. IntServ is a powerful model in which each individual application can specify to the network the quality of service it needs. QoS at the flow level implies to record in a router for all flows crossing it a state that characterizes the contracted QoS level. A reservation protocol like RSVP [5] responsible of the signaling of the QoS requirements is required. However, strict resource reservation has shown its limits for a large number of flows [6] as it requires the management of an explosive number of states in the routers.

The second proposal is the DiffServ [3] model. Unlike IntServ, in a DiffServ network, packets are not considered as belonging to a flow but as belonging to an *aggregate* of flows. They are marked as member of a known *class* of service. The class identifier [16] (DSCP, DiffServ code point), selects a specific PHB (Per-Hop Behavior) they will experience in each router they cross. Resource reservation for each individual flow along the path is not needed. Marking or admission control is expected to be done at the network boundaries. The level of guarantee provided by DiffServ is of course lower than what IntServ provides. However, the system is more scalable and can handle an arbitrary number of flows.

The DiffServ standard does not specify services that are provided in routers. Several research teams are working on the design of services for DiffServ networks. Actually two services are on the way to be standardized by the IETF: the *premium* service provided by Expedited Forwarding (EF) [15] and the *assured* service provided by the Assured Forwarding (AF) [12].

Expedited Forwarding The concept behind Expedited Forwarding (EF) [15] is to virtually separate the network in two sub-networks, the former being the best-effort network and the latter (where *premium* packets are carried) being a kind of low-latency network whose packets experience a very low loss. The premium service is particularly suitable for real-time flows since end-to-end delay is smaller than in the best-effort class. However, elastic flows obviously do obtain good performance in the premium class because of the low loss and delay.

Assured Forwarding In an Assured Forwarding (AF) [12] router, packets are separated into n queues having each one a specific share of the bandwidth. Inside a given queue, m subclasses experiment different loss rates. Even if the model does not specifies the number of classes, the service is often represented with $n = 4$ and $m = 3$. By practicing the differentiation over loss rate, assured services are not suitable to real-time flows but elastic flows.

These two propositions are based on statistical provisioning. Usage of buffer management provides a “better effort” service (AF). Premium service offers little jitter and queuing delay and does not need queue management. Premium Service has been deployed in different experimental networks like Q-Bone [20] and VTHD [24]. However, some problems are raised and still remain to be solved. For example, the Q-Bone Premium Service has not succeeded and its deployment is stopped [23].

Several others proposals for service differentiation exists (for example [8, 18, 2]). Design of new services for DiffServ is testament to the fact that QoS requirements of the Internet traffic cannot be accurately identified and no single approach is applicable and efficient in all cases. The solution spectrum is large certainly because the application and situation spectrum is large. For “business multimedia applications”, high level of quality of service is required. For many ordinary applications, the performance required can be less. One can imagine that the user will accept to pay for an important application, but not for “every day life” applications. Assuming the multimedia communication will grow up in the next future, a solution must be found to serve the various requirements of the different flows. Expedited Forwarding and Assured Forwarding PHB provide different level of quality of service what implies pricing differentiation and possibly admission control.

Within new proposals for service differentiation, which offer far weaker service assurances than usual service differentiation models is the Alternative Best-Effort (ABE) [14] model. The main goal of ABE is to provide a low-delay service in the Internet without calling for pricing differentiation or admission control. The service is thus very close to the best-effort service and easy to deploy what makes it very appealing. The low-delay class is the *green* one. Its packets are queued in a specific way that gives them a lower and bounded delay than the best-effort class (the *blue* one). However, ABE routers drop preferentially green packets. The main point in a router is that its drop differentiation ensures that the best-effort class gets at least the same performance it would in a flat best-effort network (where no delay or drop differentiation is practiced). Classes are not supposed to be priced differently and no admission control is required. Specific pricing is avoided because their is a kind of fairness in the way differentiation is practiced. Low delay class gets a higher loss in a way that gives to best-effort flows approximately the same end-to-end performance they obtain in a plain best-effort network.

Best-Effort Differentiated Service (BEDS) [9] is an other recent service model close to ABE that aims at providing the same kind of service with softer requirements. Queue management in ABE requires flows to be TCP-friendly, which is not an assumption of the BEDS model. Differentiation in BEDS is based on statistical measurements in Internet traffic and performance models of elastic and real-time applications. It provides a loss conservative queue for TCP packets and a delay conservative queue for UDP.

3 Equivalent Differentiated Services

In this section, we present the “Equivalent Differentiated Services” (EDS) model. It is similar to ABE and BEDS models in the sense it practices the same kind of trade-off between queuing delay and loss. The main difference in the EDS approach is the kind of services available in the network and the way they are used. We first remind the way the TCP/IP stack was built and show how we think the same philosophy can be applied to a QoS network.

3.1 IP Philosophy and QoS

Packet forwarding in IP is a very simple model which simplicity and robustness is one of the key of its great deployment. Two main points of the model is that the use of the network layer is free from specific pricing since the entire communication resource is available for all users. No admission control is practiced. Free from admission control, the network can obviously be noised by unfriendly users. However, this makes the system very easy to implement and deploy. Even if ABE and BEDS models add QoS to the network, they do respect these characteristics: no specific pricing nor admission control are required.

They however differ from the IP model on the following point. In plain IP, router behavior takes into account only local criteria (like queue capacity in TailDrop or average queue length in RED). Their setting is not based on the kind of traffic or the protocols that cross them. Best-effort is a simple local *building-block* that end-to-end protocols freely use to provide their services. ABE and BEDS made the choice of providing "end-to-end protocols-aware" services at the IP level. The first ABE proposed implementation [13] used to be bound to the TCP performance model. The last known implementation [14] is less strict but the model needs flows to be TCP-friendly. In BEDS, setting is based on performance models of end-to-end flows.

The direction we want to explore is the definition of "end-to-end unaware" QoS services that matches all three criteria defined above. The key point being that the network layer does provide services, but it is not defined under any existent transport layer. These services are similar to the best-effort service because they are provided as they are to end-to-end protocols. As we will see in sect. 3.3, this choice makes setting a different task than in other "different but equal" models.

We previously designed a model called "Balanced Forwarding" (BF) [11]. The model conforms to the first two points. To avoid a network layer bound to the global context, queue management depends only on local values so that the model is conform to the third point. However, the parameters setting is a complex problem due to its dynamic behavior and requires studies.

3.2 The EDS Model

The EDS model is defined over an arbitrary number $N \geq 2$ classes. Differentiation is practiced over delay and loss rate of each class. A class i gets a delay coefficient d_i and a loss rate coefficient l_i. These coefficients are constant. Their meaning is the following: let i and j be two classes, the router schedules and drops their packets so that their is a ratio d_i/d_j between their *local* queuing delays and l_i/l_j between their *local* loss rates. To avoid having privileged classes, coefficients are set so that if $d_i < d_j$ then, $l_i > l_j$ and vice-versa. On fig. 1, the eight available classes do have different performances in delay and loss rate, but asymmetry is ensured in coefficient setting.

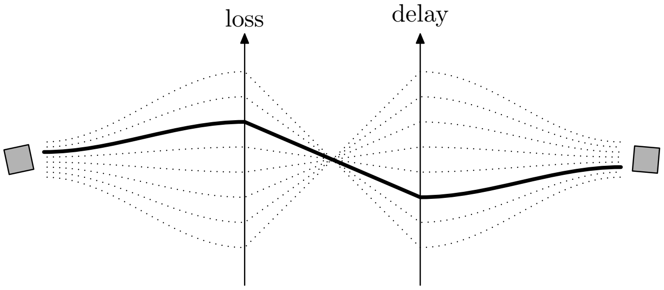

Figure 1: The EDS model set with eight classes. The packet class identifier corresponds to a class which loss rate is higher than the average loss rate in the router. However, its queuing delay is lower.

3.3 Parameters Setting

As we said above, setting of l_i and d_i parameters is the main different point of EDS. Others "different but equal" models take into account flow characteristics or statistics (the network layer is built under the existent transport layer). In the EDS model, setting is an open problem since it is not based on the transport layer.

This is similar to queue capacity or RED thresholds setting. However, as it is the case on top of the best-effort service, the transport layer is expected to build its services on top of these new services.

As an example, fig. 2 represents how differentiation could be configured in an EDS router. The first two (leftmost) configurations are similar to the one shown on fig. 1 in the sense delays and loss rates are regularly spread on the diagram. The last two configurations provide two main categories of classes where delay and loss rate differentiation is not practiced the same way. For example, on the first one, high delay classes experience a high loss rate differentiation (the range of l_i is greater than the range of d_i) and vice-versa. EDS routers configuration depends on the kind of network services that would be relevant to the traffics that are carried. Then, protocols are ported on top of them. A network of routers practicing these kinds of differentiation provides to a flow a set of different packet forwarding services corresponding to different classes. The flow chooses a class by marking the class identifier of its packets. When the network load increases or decreases, the packet forwarding performance is affected and may not match the flow criteria. The flow can then switch to an other class if the new one may fill its needs better. To get full benefit of the EDS services, the packet marking has to be made at the end nodes. An example of the use of EDS for real-time traffic is presented in sect. 4. The case of elastic traffic is more complicated and is object of ongoing studies (see sect. 5 for a short description of these studies).

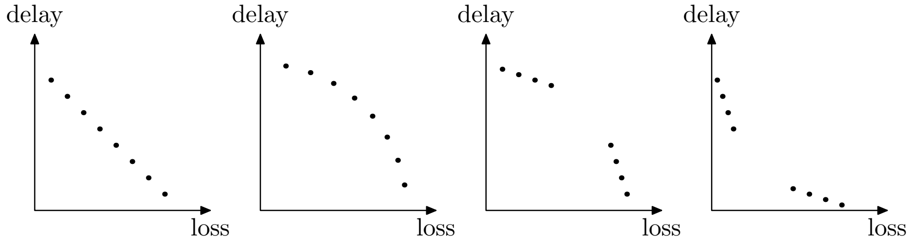

Figure 2: Example of configurations of an EDS router.

3.4 A scheduler for EDS

In this section, we introduce a scheduler that implements the EDS PHB. In EDS, differentiation is practiced over two statistics: queuing delay and loss rate. Since differentiation is proportional, the scheduler is inspired by those presented in [8]: Waiting-Time Priority (WTP) and Proportional Loss Rate dropper (PLR).

Waiting-Time Priority The WTP scheduler practices a proportional differentiation on queuing delay. Each class i owns a specific queue that is given a coefficient d_i. Each time a packet is enqueued, its arrival date is recorded. When selecting the packet that must be dequeued, the scheduler computes the virtual waiting-time $\bar{\omega_i} = d_i\omega_i$ of the first packet of each queue where ω_i is the waiting-time of the packet. The higher d_i, the higher $\bar{\omega_i}$. The scheduler selects the packet whose virtual waiting-time is the highest.

Proportional Loss Rate dropper The PLR scheduler is used to practice a proportional differentiation on loss rate. Each class i is given a coefficient l_i. Packets are queued in a common FIFO queue, whatever class they belong to. The scheduler maintains a history table of the K last received packets, that permits to compute the average loss rate ρ_i of each class. As in WTP, loss rates are normalized to compute the virtual loss rate $\bar{\rho_i} = l_i\rho_i$ of each class. The higher l_i, the higher the virtual loss rate. When the router capacity is reached, instead of dropping new packets, the router selects the class that has to lose a packet: the one whose $\bar{\rho_i}$ is the lower.

The scheduler we designed for EDS is shown on fig. 3. It is a mix between WTP and PLR. A new packet is first recorded in the history table and gets the current date. Moreover, it gets a memory reference to the history table cell that contains its information. Since the scheduler may later choose it to drop it, the corresponding cell can be efficiently found and updated. The packet is enqueued according to its class identifier and waits either for the WTP to dequeue it or for the PLR dropper to drop it. We did not study the algorithm complexity. Computation of virtual delays is the most expensive task (n computation where n is the number of classes). The entire algorithm complexity seems to be less than WFQ complexity.

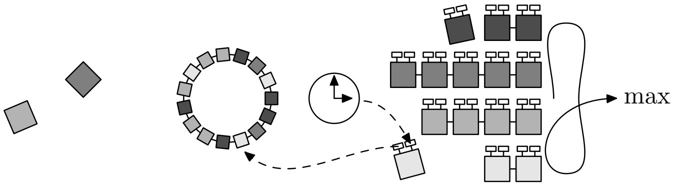

Figure 3: A scheduler for EDS.

4 Experimental Study

In this section, utilization of EDS services is demonstrated in a simulated experiment on a EDS network. The scheduling discipline was implemented in the Network Simulator (NS) [17]. It was written upon Dovrolis' WTP implementation and we checked our implementation of PLR was correct.

We study the case of a flow with strong delay requirements (say a real-time flow) which packets needs to be carried in less than 150 ms. However, it shares the network with concurrent traffic that causes annoying end-to-end delay variations. The EDS system gives to the flow a kind of control over the end-to-end delay. It uses this functionality to obtain a more satisfying QoS. We first give the broad outlines of the adaptive algorithm, then we study its performances.

4.1 Flow adaptation principles

Flow constraints are that emitted packets have to be received in less than $t < \Delta_M$ because data received after Δ_M are not used by the application.

For each packet the destination receives, it computes the transmission delay. Thanks to this value, it computes the average delay $\bar{\Delta}$. When it notices that $\bar{\Delta} > \Delta_M$, most of the packets are uselessly carried so that the average transmission time should be decreased.

The receiver sends then a control message to the source that indicates the new class identifier which delay is smaller. When $\bar{\Delta}$ is greater but close to Δ_M, the receiver selects an adjacent class, if the difference is large, it chooses a more distant value. When it receives the message, the source just resets the class ID with the new one. After having sent a control message, the receiver still computes the average delay during $2\bar{\Delta}$ before comparing its value to Δ_M and having an idea of the performance of the new class.

The receiver could systematically select the lower available latency that however gets the higher loss. To avoid that case, it chooses a class with higher delays if $\bar{\Delta} \ll \Delta_M$. Then, latency may increase again but not necessarily reaching the delay-bound while loss may decrease anyway. To avoid oscillations of delay around Δ_M, the receiver reacts sooner, when $\bar{\Delta}$ is close to Δ_M.

If using the best low delay class (which delay is the highest below Δ_M), the loss rate due to loss differentiation is too high, the network cannot fit quality of service requirements of the application.

4.2 Experiment

In this section, we study the performance of the real-time flow injected in a busy network, crossed by non-deterministic (TCP) flows. The flow sends packets at a 64 Kb/s rate and expects a delay bound $\Delta_M = 150$ ms. The experimental network is shown on fig. 4. Router/router links have a bandwidth of 4 Mb/s while node/router links have a bandwidth of 30 Mb/s. Links latency is 5 ms. EDS routers are set up with 16 classes which coefficients d_i and l_i are set so that differentiation is similar to the leftmost one shown in fig. 2. On fig. 4 the delay-constrained flow is represented by the $N_2 \rightarrow N_6$ arrow. Arrows $N_1 \rightarrow N_5$ and $N_3 \rightarrow N_4$ represent two sets of concurrent flows generated by TCP. An arrow stands for 64 connections. For each class of service, $64/16 = 4$ TCP connections use it in $N_1 \rightarrow N_5$ and $N_3 \rightarrow N_4$. $N_1 \rightarrow N_5$ connections run during the entire experiment. $N_3 \rightarrow N_4$ connections start in the middle of the experiment and stop at the end. Fig. 5 shows the end-to-end latency obtained by packets of the application during a 30 s long experiment. With best-effort routers (see fig. 5 on the left), the main point is that in the middle of the experiment, the delay becomes really high because of the $N_3 \rightarrow N_4$ connections while the application cannot control it. The graph for EDS routers (fig. 5 on the right) shows at the same time the evolution of the end-to-end delay and the class ID. In this experiment, the concurrent traffic is slightly different to the one experienced in the best-effort experiment since differentiation

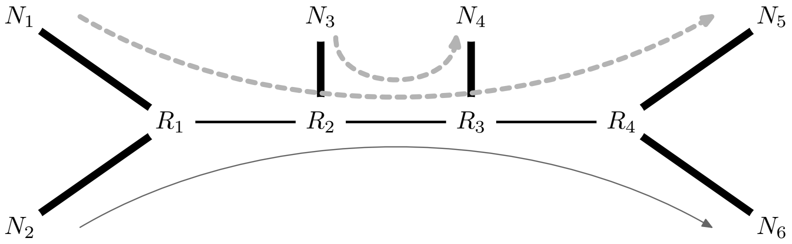

Figure 4: The experimental network.

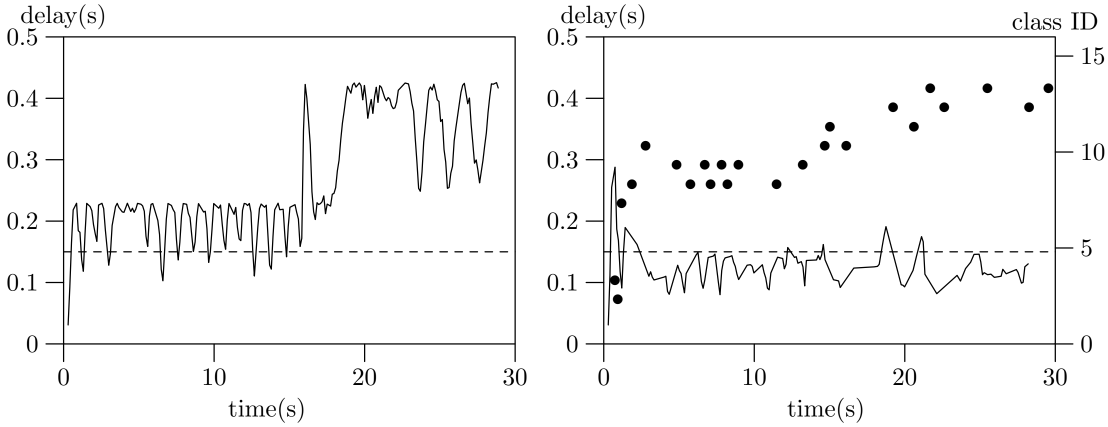

Figure 5: Latency obtained by packets of the real-time application on top of best-effort (left) and EDS (right). Rightmost figure also shows the evolution of the class ID along the experiment.

takes place. However, the evolution of network load is similar. At the very beginning, the flow uses a class which delay is obviously too large (see the peak). It quickly moves to a more adapted class ID around 8 and 9. When $N_3 \rightarrow N_4$ connections start, the latency increases in a similar way to the best-effort experiment. The application moves to a higher class ID with lower queuing delays (around 13 and 14).

5 Conclusion

We have presented the "Equivalent Differentiated Services" building block for a multi-service Internet which is simple, robust and scalable. The EDS mechanisms can be partially deployed. In an overprovisioned backbone, traffic does not need any specific treatment. The EDS can be deployed in access networks were congestion occurs more frequently. The model is similar to existent "different but equal" models but its originality lies in the fact that differentiation is not bound to the existent traffic models or transport layers. Services are provided as they are and end-to-end protocols are built on top of them. The network architecture is thus very close to the way the TCP/IP stack was built.

Services provided by end-to-end protocols or applications that use EDS classes efficiently should be better than on top of best-effort because of the several available services. We have presented the case of a real-time application that uses the delay differentiation to ensure a transmission time below a given threshold. The case of elastic traffic is what our current work mainly focuses on today. Building a reliable service like TCP on top of EDS is a challenging task. It is conceivable to adapt TCP to EDS by making it use the best available class in trade-off between loss and delay. A more satisfying adaptation would be to make TCP use different classes during slow-start, for acknowledgments or retransmission of lost packets if it is shown to make the goodput better. Moreover, the same reliable end-to-end service could be completely redesigned and made configurable according to the delay requirements of the data (e-mail *vs.* telnet) for example. We are exploring these three directions today. Thus, EDS provides a support for end-to-end services to be better, but also can make them become *richer* and match the needs a large range of applications.

References

[1] Audio-Video Transport Working Group, Henning Schulzrinne, Steve Casner, Ron Frederick, and Van Jacobson. RTP: A transport protocol for real-time applications. Internet Request For Comments RFC 1889, Internet Engineering Task Force, January 1996.

[2] Albert Banchs and Robert Denda. A scalable share differentiation architecture for elastic and real-time traffic. In *Proceedings of International Workshop on Quality of Service (IWQoS)'00*, pages 42–51, Pittsburgh, PA USA, June 2000. IEEE/IFIP.

[3] 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.

[4] Robert Braden, David Clark, and Scott Shenker. Integrated services in the internet architecture: an overview. Internet Request For Comments RFC 1633, Internet Engineering Task Force, June 1994.

[5] Robert Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin. Resource reservation protocol (RSVP) – version 1 functional specification. Internet Request For Comments RFC 2205, Internet Engineering Task Force, September 1997.

[6] Tzi-cker Chiueh, Anindya Neogi, and Paul Stirpe. Performance analysis of an RSVP-capable router. *IEEE Network*, 13(5):56–63, September 1999.

[7] David D. Clark. The design philosophy of the DARPA internet protocols. In *Proceedings of Special Interest Group on data Communication (SIGCOMM)'88*, number 4 in Computer Communication Review, pages 106–114, Stanford, CA USA, August 1988. ACM, ACM Press.

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

[9] Victor Firoiu and Xiaohui Zhang. Best effort differentiated services: Trade-off service differentiation for elastic applications. In *Proceedings of International Conference on Telecommunications (ICT)'01*, Bucharest, Romania, June 2001. IEEE.

[10] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[11] Benjamin Gaidioz, Pascale Primet, and Bernard Tourancheau. Differentiated fairness: Model and implementation. In *Proceedings of High Performance Switching and Routing (HPSR) 2001*, pages 260–264, Dallas, USA, May 2001. IEEE.

[12] Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski. Assured forwarding PHB group. Internet Request For Comments RFC 2597, Internet Engineering Task Force, June 1999.

[13] Paul Hurley and Jean-Yves Le Boudec. A proposal for an asymmetric best-effort service. In *Proceedings of International Workshop on Quality of Service (IWQoS)'99*, pages 132–134, London, England, June 1999. IEEE/IFIP.

[14] Paul Hurley, Jean-Yves Le Boudec, Patrick Thiran, and Mourad Kara. ABE: Providing a low-delay service within best effort. *IEEE Network*, 15(5), May 2001.

[15] Van Jacobson, Kathleen Nichols, and Kedar Poduri. An expedited forwarding phb. Internet Request For Comments RFC 2598, Internet Engineering Task Force, June 1999.

[16] Kathleen Nichols, Steven Blake, Fred Baker, and David Black. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. Internet Request For Comments RFC 2474, Internet Engineering Task Force, December 1998.

[17] Network simulator ns-2 web site. `http://www.isi.edu/nsnam/ns/`.

[18] Andrew Odlyzko. Paris metro pricing: The minimalist differentiated services solution. In *Proceedings of International Workshop on Quality of Service (IWQoS)'99*, pages 159–161, London, England, June 1999. IEEE/IFIP.

[19] Jon Postel. Transmission control protocol. Internet Request For Comments RFC 793, Internet Engineering Task Force, September 1981.

[20] Internet2 Q-Bone web site. `http://qbone.internet2.edu`.

[21] K. K. Ramakrishnan and Sally Floyd. A proposal to add explicit congestion notification (ECN) to IP. Internet Request For Comments RFC 2481, Internet Engineering Task Force, January 1999.

[22] Scott Shenker, Craig Partridge, and Roch Guerin. Specification of guaranteed quality of service. Internet Request For Comments RFC 2212, Internet Engineering Task Force, September 1997.

[23] Benjamin Teitelbaum. Future priorities for Internet2 QoS, October 2001. `http://qbone.internet2.edu/qos/wg/papers/qosFuture01.pdf`.

[24] VTHD web site. `http://www.vthd.org`.

[25] John Wroclawski. Specification of the controlled-load network element service. Internet Request For Comments RFC 2211, Internet Engineering Task Force, September 1997.

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