Quality of Services for Remote Control in the High Energy Physics Experiments: a Case Study

T. Ferrari, A.Ghiselli, C.Vistoli
Italian National Institute for Nuclear Physics, CNAF

The development of new advanced applications and the evolution in networking are two related processes which greatly benefit from two-way exchanges and from progress in both fields. In this study we show how mission-oriented networked applications can be effectively deployed for research purposes if coupled to the support of Quality of Service (QoS) in IP networks. QoS is one of the latest research topics in network engineering.
In this article [1] we focus on two specific examples of networked applications: remote instrumentation control and remote display of analysis data when applied for the support of experiments in the high energy physics field. In this paper we focus on the application requirements: the availability of a reliable transmission channel, limited one-way delay for timely interactions between servers and clients and fairness in network resources allocation in case of contention.
The above-mentioned requirements can be addressed through the support of QoS, i.e. through the differential treatment of packets on the end-to-end data path.
Several technologies and protocols for QoS support in packet networks have been devised during the last years by the research community. In this study we focus on the Differentiated Services (diffserv) approach, an architecture characterised by high scalability, flexibility and interoperability.
In this paper we identify the application requirements and we quantitatively specify the corresponding service profiles. The diffserv network architecture needed to support the services is defined in terms of functional blocks policing, classification, marking and scheduling) and of their placement in the network. Finally, for each of them the configuration best suited to remote control support is defined.

Keywords: Quality of Service, Differentiated Services, remote control, CDF
PACS code: 84.40.Ua

[1] This work has been carried out in the framework of the INFN project QUAdiS.

1 Introduction

Remote control is a functionality with many application cases in several different research fields. In this paper we address the problem of remote instrumentation control when applied to the high energy physics, in particular to CDF experiment: *the Collider Detector at Fermilab* [1].
Remote control addresses the problem of the effective participation to international high energy physics experiments by researchers located far from the laboratories. The support of reliable, efficient and guaranteed network transactions in remote instrument control through Quality of Service (QoS) mechanisms is the aim of this study.
In CDF remote control requires two main functionalities: the handling of the trigger hardware configuration from a remote control site and the monitoring of the analysis through the remote display of results in multiple client sites. The *Rpc and Object Broker INterface* (ROBIN) [2] is the software platform which provides support to the former task, while ROOT, the *Object Oriented Data Analysis Framework* [3] is the tool used for the display of results in remote monitoring sites.

2 QoS: the differentiated services architecture

Classification consists in the differentiation of traffic classes through the association of an identifier to a set of one or more data flows. In the differentiated services architecture [4] two packet classification approaches can be adopted:

- *Multifield Classification*: packets are identified through the content of a combination of IP header fields like the source or destination IP address, the port number and the protocol type.
- *DSCP-based Classification*: packets are associated to the class they belong to through the so-called *Differentiated Services Code Point* (DSCP) [6] in the IP header. The DSCP can be set by the application and/or by the edge router of the first diffserv domain encountered on the data path to the destination.

The adoption of *admission control* [7,8] is recommended at the edge of the network to guarantee that the proper binding between packets and labels is in use, i.e. to verify that only applications which are entitled to do so actually mark packets by placing the right DSCP into the IP header. Admission control is implemented according to configurable *policies*.

Marking consists in the placing of a code point into the IP header of a packet for the identification of the class that packet belongs to. Marking is normally performed at the edge of a diffserv domain: In this way only edge routers need to keep the information for the mapping of flows or microflows to diffserv

classes and to maintain policies. For scalability sake core diffserv nodes trust the marking performed at the edge and have no knowledge of the identity of single flows: instead of providing per-flow QoS guarantees, internal diffserv nodes only handle flows aggregations, i.e. they do not make any distinction between packets in the same class. The DSCP of a given packet can be changed on the way to the destination, if this occurs, than we say that the packet is subject to *re-marking*. Re-marking can occur multiple times at boundaries between two distinct diffserv regions when the same kind of packet treatment is identified differently in contiguous domains.

3 Application description and characterisation

Whatever technology is deployed for QoS support, the application needs to be characterised to identify the most suitable service, which can be defined either qualitatively or quantitatively. The sensitiveness to delay and/or jitter, the tolerance to packet loss, the traffic volume issued by the application, the burstiness degree of the sources, the presence of potential traffic collision points in the networks, are some of the factors which need to be taken into consideration. Data packets issued by applications need to be identified and classified.

3.1 Hardware remote control

VME crates are controlled through a server hosted by a PowerPC. Multiple servers are present at a time, one for each crate, while up to n distinct remote clients can be active simultaneously, as illustrated in Fig. 1.
As the scheme shows, potential collision points are the links connecting clients to the networking infrastructure - in case of simultaneous access to multiple server - and the whole end-to-end data path in case of background traffic competing for resources. One server interfaces only one client at a time, while a client can control multiple VME crates in parallel. Hardware control is performed through commands which invoke specific routines to access the VME crate.
If control is performed over a best-effort network, then packets may suffer from long- or short-term congestion on the links providing access to the network infrastructure in case of simultaneous access to multiple server or in presence of background general-purpose traffic.
Remote control is performed through the *bidirectional* exchange of IP datagrams between one server and the corresponding client. The estimated traffic volume produced by a single session is limited Hardware control is scarcely interactive since only small information items (commands) are set out over the network. The frame payload size of a packet is less than 100 bytes and

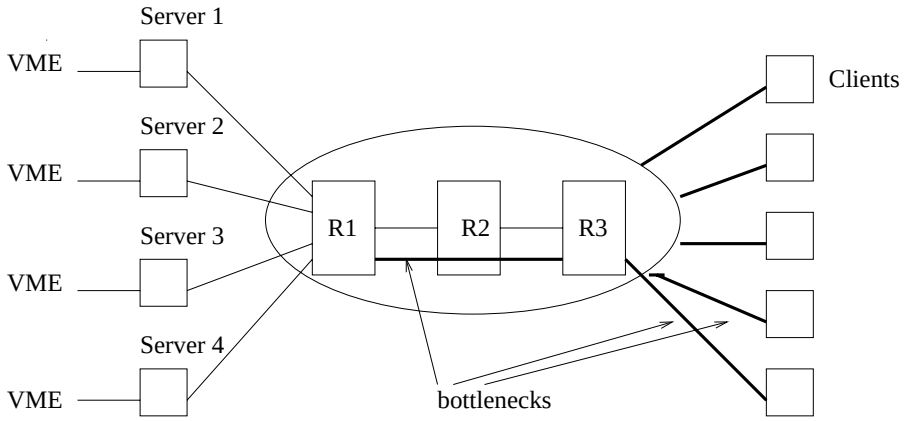

Fig. 1. server-client interaction scheme for remote control of VME crates

burstiness is negligible.

IP frames are transported through the TCP [9] reliable transport protocol and both clients and servers are identified by a known IP address. In addition, the server port number is fixed: 50038, while the client port number can vary. As a consequence, packet marking can be easily implemented through multifield classification, i.e. through the the source/destination address, the protocol type TCP and the source/destination port number.

Traffic is delay bound since the effect of commands on the trigger hardware should be as immediate as possible. In addition, packet loss under TCP implies the retransmission of datagrams, as a consequence this application requires high reliability during transmission.

3.2 Monitoring of the analysis

Monitoring of the data analysis requires an interactive access to the results: Raw data analysis is performed centrally and a single server [2] manages the access to the local repository. Multiple clients can issue requests to the server in parallel, as illustrated in Fig. 2. For this application the potential bottlenecks are the access link connecting the server to the network in both ways - in particular in the server-to-client direction - and the whole end-to-end path, especially the edge routers on the client side in the server-to-client direction if a single monitoring site hosts multiple clients. Results are sent out to clients in table format displayed by ROOT remotely in monitoring site.

Remote control and remote analysis are two related applications, since depending on the analysis output one client can trigger a new remote instrumentation

[2] The database architecture to be deployed in the CDF experiment is under definition at the time of writing. In this paper we stick with the simplest scenario, in which a single database managed by a unique server is adopted.

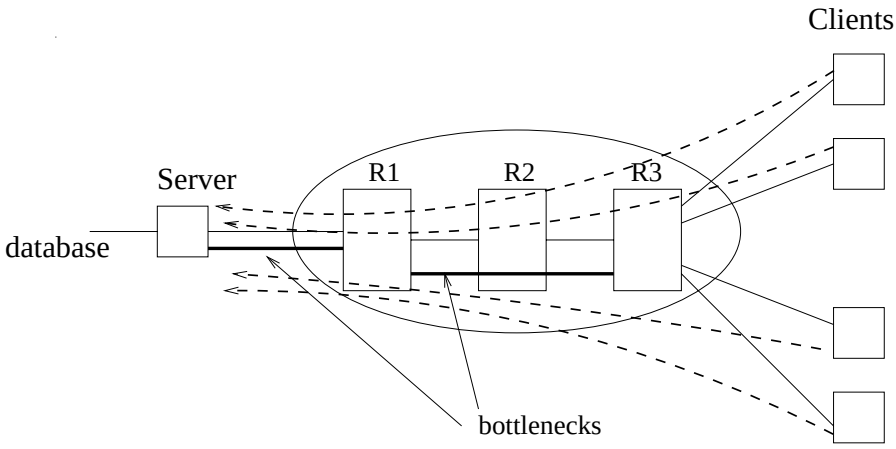

Fig. 2. server-client interaction scheme for remote analysis

control transaction.

Also in this case the data exchange is bidirectional, but the traffic volume is asymmetric as most of the data is transmitted in the server-to-client direction. Data is exchanged so that limited amounts of data (ROOT objects) are exchanged for the display of analysis results in remote sites and burstiness is negligible. Both server and clients are identified by well-known IP addresses and traffic deploys the TCP transport protocol.

This application is packet loss tolerant, however since it is also interactive, in case of congestion a minimum amount of bandwidth needs to be guaranteed. However, monitoring is not either one-way delay or jitter-sensitive.

The overall traffic volume issued by the server should be provided with a maximum bandwidth guarantee equal to threshold max, while exceeding traffic can be either treated as best-effort or dropped. In addition, traffic to a given client has to be protected from misbehaving users by guaranteeing a minimum per-user bandwidth bw_i. However, each client should be given the possibility to deploy up to the maximum overall capacity max if not allocated to other monitoring agents.

4 Service level specifications

According to the previous application characterisations the corresponding services are defined in section 4.1 and 4.2 respectively. In what follows services will be referenced with the term *Service Level Specification* (SLS).

- Bandwidth guarantee: only if traffic is below an upper threshold bw, e.g. 512 Kbps, packets are marked with high-priority: excess traffic is transmitted in a best-effort fashion [3]. Given routers $R1$ and $R3$ of Fig. 1, the value of parameter bw can be estimated according to formula:

$$bw = x * 64Kbps$$

  where x is the number of servers which can be accessed through router $R1$ or of clients connected by router $R2$.
- Burst tolerance: any packet belonging to a burst whose size is smaller than the upper threshold $B = 64\,KBytes$ is marked with a high-priority label, otherwise the packet is treated as best-effort [4].
- Delay bound: one-way delay is upper-bounded. Our definition of this metric is according to the one specified in RFC 2679 and corresponds to metric `Type-P-One-way-Delay`. Given a source Src and a destination Dst one-way delay is the time elapsed from the time at which a bit of Type P is issued by Src to the time at which the last bit of that packet is received by Dst. The upper delay bound D expressed in msec can be estimated through the formula:

$$D = \frac{1}{2}RTT + x * 10\,msec$$

  where RTT is the round trip time of a packet of 100 bytes, x the number of routers on the data path and 10 msec is an approximation of the maximum nodal delay introduced by a diffserv router using a transmission queue serviced according to the FIFO policy [11] [5].

The service specification defined above can be deployed in both directions, i.e. from server to client and vice versa. The value of parameters defining tolerances, like the bandwidth guarantee, burst tolerance and the delay bound can be tuned according to the traffic volume in each direction.

4.2 Service 2: monitoring of the analysis

While in the previous case a unique service can satisfy the application requirements in both directions, given the asymmetry of the two data streams, two

[3] 512 Kbps is a reference value, a different and more appropriate bound can be chosen depending on the number of local clients connected to the edge router.

[4] The maximum buffer size B can be tuned as needed. The optimum value can depend on the instantaneous traffic volume, i.e. on the number of local servers or clients.

[5] By picking 1028 bytes we get a worst-case estimation of the nodal delay, since in production networks the average datagram size is in the range [300, 400] bytes.

different services have to be defined.

4.2.1 Client → Server SLS

- Bandwidth: each client is guaranteed with a minimum rate of 64 Kbps to the server [6], nevertheless it is allowed to generate a traffic volume up to 256 Kbps (or an equivalent value higher than 64 kbps) in case of resource availability.

  Packets for which the instantaneous traffic rate exceeds the upper rate threshold are dropped Alternatively, excess packets can be treated with a higher dropped priority or they can be provided with a best-effort service. The drawback of the second approach is that packets belonging to the same microflow but marked as best-effort need to be separated into two different queues with consequent packet reordering. On the other hand, if exceeding traffic is marked with a non-be-effort label, they can still placed in the same queue: traffic differentiation is achieved through the Weighted Random Early Detection [12,13] packet discard mechanism according to which packets exceeding the lower threshold (64 Kbps) are dropped first.
- Burst tolerance: 16 KBytes (or an equivalent value defined through tuning).

4.2.2 Server → Client SLS

- Bandwidth: for each client the server can deploy 64 Kbps of guaranteed bandwidth. This means that the overall amount of guaranteed bandwidth is equal to $64Kbps*m$ where m is the number of clients (if for each client the amount of guaranteed bandwidth $bw(i)$ is the same). 64 Kbps is a reference value which can be appropriately tuned.

  If plenty of bandwidth is available, the overall rate of traffic generated by the server can be up to 5 Mbps, this means that if some clients are not active at a given time, a given client can deploy from 64 Kbps to 5 Mbps. Traffic exceeding 5 Mbps is dropped: in case of congestion packets exceeding the minimum per-client guaranteed bandwidth (64 Kbps) are dropped first.
- Burst tolerance: traffic bursts up to 128 KBytes are tolerated. This reference value can be tuned and modified appropriately, if needed.

5 Services Implementation

5.1 Service 1

[6] 64 Kbps is a reference value which needs to be tuned appropriately.

PHB Given the delay sensitiveness and the limited traffic volume in remote hardware control, the most suitable Per Hop Behaviour is the Expedited Forwarding (EF) PHB [14].
In the discussion which follows we assume that traffic exceeding a given rate threshold receives a best-effort treatment instead of being dropped.
Conforming packets are marked with DS codepoint 101110 (the standard EF codepoint). In case of diffserv nodes which only support Class Selector codepoints [4] then precedence value 5 (101) is used. In case of heterogeneous domains with both DSCP and Class Selector capable diffserv nodes, re-marking has to be deployed within the domain.

Marking Marking is deployed at the edge of the diffserv domain, i.e. at routers $R1$ and $R2$ in Fig. 1, where $R1$ represents an edge router connecting one or more servers to the diffserv domain and $R2$ is a generic edge router with one or more clients located downstream. The server edge router ($R1$) marks packets in the server-to-client direction, while the client edge router ($R2$) marks packets in the opposite one according to the following scheme.
Given the IP address of server i $IP_s(i)$, the IP address of client j $IP_c(j)$ and a packet p, marking is implemented according to the following rule:

```
Server edge router:
if ((dest-IP-add(p) in {IP_c(1), IP_c(2), ... , IP_c(n)})
     && (src-IP-add(p) in {IP_s(1), IP_s(2), ... , IP_s(m)})
     && (src-port(p) = 50038) && (protocol(p) = TCP))
then
     DSCP = EF

Client edge router:
if ((dest-IP-add(p) in {IP_s(1), IP_s(2), ... , IP_s(m)})
     && (src-IP-add(p) in {IP_c(1), IP_c(2), ... , IP_c(n)})
     && (dest-port(p) = 50038) && (protocol(p) = TCP))
then
     DSCP = EF
```

Policing Policing is enabled at the edge of the network to guarantee that EF traffic does not exceed a given upper threshold. Policing can also be adopted at the boundary between different diffserv domains in case of a diffserv region composed of multiple DS domains. In this case scheduling may be performed on the data path by the last router belonging to a given domain to guarantee that the instantaneous EF rate does not exceed the bandwidth upper bound specified in the contract.
We assume that policing is implemented through a multi-parameter token bucket [5] deploying both a *normal* and *excess* burst size ($norm_b$ and $excess_b$ respectively) for traffic metering:

```
if (norm_b = B && excess_b = B && rate <= bw) then
        DSCP = EF
else
        DSCP = best-effort
```

Not only value *bw* depends on the traffic issued by our application, but also on the presence of additional microflows generated by independent applications which may be merged into the same class. In this case, all the three token bucket parameters need to be tuned to take into consideration the requirements of each application.
Policing has to configured in each edge router in the diffserv domain connecting one or more server and/or clients to the infrastructure.

Scheduling Scheduling is the mechanism used to define the order of service of packets stored in different queues [17]. In order to minimise nodal delay priority queueing [18,19] is recommended as EF service policy. Priority queueing guarantees that if at a given instant t the priority queue is not empty, then the scheduler selects the packet at the head of the priority queue for transmission.

A priority queue has higher precedence than any other queue in the scheduling system, as a consequence an upper bound to priority traffic is always recommended to avoid starvation of other queues.
The maximum queueing time introduced by a priority queue is equal to the transmission time of an MTU-size packet. Additional delay may be introduced by the transmission queue, if it is FIFO and it already stores one or more lower-priority packets when the PQ packet arrives. [20] provides an in-depth analysis of priority queues in terms of one-way delay and jitter.
Experimental and simulative studies compare priority queueing with other scheduling algorithms like Weighted Round Robin (WRR) [14] and Weighted Fair Queueing [17,11]. Priority queueing is indicated as as the most appropriate queueing algorithm for the optimisation of one-way delay. The gain can vary depending on ratio between the average EF packet size and the average best-effort packet size as illustrated in Fig. 3.

Fig. 3 reports on the experimental results obtained when comparing priority queueing with WFQ. In the two cases average one-way delay is measured for different expedited forwarding packet sizes. The network interface to which scheduling is applied is artificially congested through a background best-effort stream competing against the expedited forwarding stream. The best-effort IP packet size is constant and equal to 1028 bytes.
For example, if the the EF packet size is small (e.g. below 512 bytes), then the performance of WFQ and PQ are approximately equivalent. On the other hand, for large EF packets PQ performs much better.
This behaviour is expected: The transmission time of a packet is a function of both the weight of the queue it belongs to, and of its size. As a consequence,

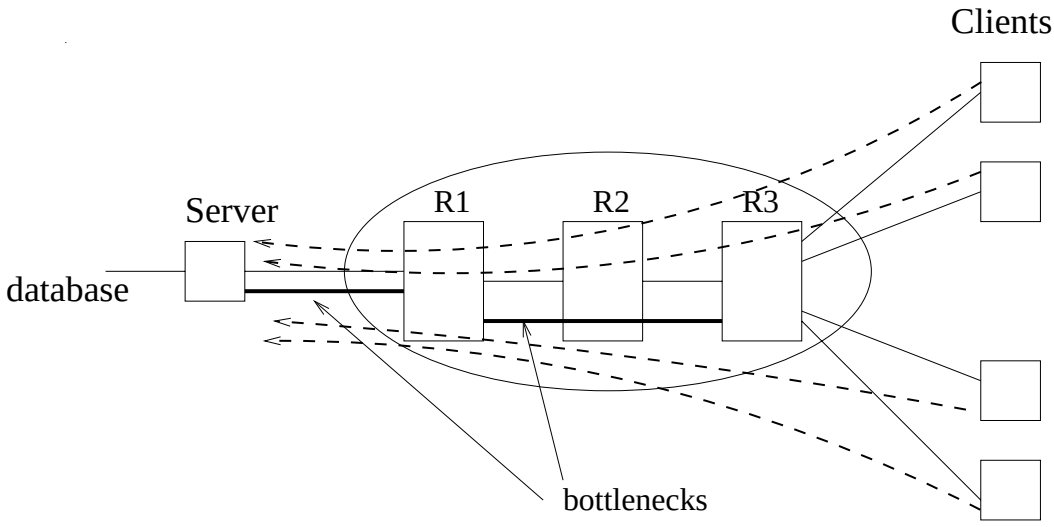

Fig. 3. comparison between priority queueing and weighted fair queueing in terms of average one-way delay for different Expedited Forwarding packet sizes (EF load is 300 Kbps).

for small sizes - small in comparison with the size of packets in other queues - the packet delivery is more timely and WFQ converges to PQ.

Core diffserv router The core diffserv network can be configured according to two different approaches:

- Bandwidth over-provisioning is adopted in the core. This means that low probability of instantaneous congestion is guaranteed by over-estimating the available line rate in relation with the overall traffic volume. No traffic differentiation is supported, as a consequence, in case of instantaneous bursts in some behaviour aggregates, expedited forwarding packets may suffer from high queueing time.
- For each traffic direction differentiation is enabled by configuring scheduling in each egress interface of a diffserv node which is crossed by expedited forwarding datagrams. Queueing time is upper bounded in each node and reliability is achieved through the protection of expedited forwarding packets from concurrent behaviour aggregates.

5.2 Service 2

PHB For services in both directions the Assured Forwarding PHB Class can be deployed [15]. According to the standard, for each AF PHB Class, up to 3 PHBs can be deployed, so that three different packet treatments are supported within a single class depending on the DSCP of a packet. One of four possible AF classes can be selected. Since AF classes are equivalent, any can be chosen,

in the following we suppose that AF class 1 is deployed.

Within a given class the DSCP defines the drop precedence of the datagram. In our study only two levels are needed: The transmission of packets with instantaneous rate below a given threshold are provided with the lowest drop precedence (codepoint AF11), while packets whose instantaneous rate is between the above threshold and a given maximum are preferentially dropped in case of congestion (codepoint AF12).

Marking and policing For both services a kind of colour-blind Two Rate Three Color Marker can be adopted according to RFC 2698. However in this case only a single rate CBR (Committed Bit Rate) and a single burst size CBS (Committed Burst Size) are used for classification an marking.
Since we need to support two independent services in two different directions, the same PHB class AF1 can be independently deployed at the same time in the two directions.
Marking can be implemented according to the following scheme:

```
norm_b = excess_b \* normal and excess burst sizes are the same *\
if (rate <= bw && burst <= norm-b) then
        DSCP = AF11
else {  DSCP = AF12;
        if (rate(AF12) <= max && burst <= B) then transmit;
        else drop;}
```

Classification While in the client-to-server direction any packet can directly put into the same class (AF PHB class) – the actual PHB is defined by the two rate colour marker –, in the server-to-client direction a distinct traffic classifier can be configured for each client. In this way fairness is enforced by protecting traffic from misbehaving agents.

Given the server IP address of the server IP_s, of the i^{th} client $IP_c(i)$ and a packet p, traffic filters can be configured according to the following model:

```
From client to server:
   dest-IP-add(p) = IP_s && (src-IP-add(p)=IP_c(1) || ... || IP_c(m))

From server to client:
   filter_1 : dest-IP-add(p) = IP_c(1) && src-IP-add(p) = IP_s
        ...
        filter_m : dest-IP-add(p) = IP_c(m) && src-IP-add(p) = IP_s
```

Scheduling In both services packets marked with PHB AF11 and AF12 need to be placed in the same queue in order to avoid re-ordering. WFQ is the most appropriate scheduling algorithm, since for each queue a minimum

bandwidth can be guaranteed, but more capacity can be allocated to the queue if spare bandwidth is available. In this way the actual output rate provided to AF class 1 can oscillate between the minimum bw and the maximum max as specified in paragraph 5.2. More than max bps cannot be allocated because policing limits the rate to that threshold.

With WFQ the service rate of a queue defines the weight of the queue itself, a parameter which determines its preference level. The service rate corresponds to the minimum bandwidth the queue is provided with. In this case study for both directions it corresponds to the minimum bandwidth allocated to the AF class, i.e.

- from client to server: $i * bw$, where i is the number of clients connected by the edge router to the diffserv network;
- from server to client: $m * bw$, where m is the number of monitoring agents.

The AF1 queue length can be set to a large value, since AF traffic is not sensitive to queueing delay. A reference value can be 64 packets.
In order to provide AF11 and AF12 packets with a different treatment, WRED has to be enabled in the AF1 queue.
Traffic differentiation with WRED is effective only when applied to TCP traffic. With RED the average queue size is constantly monitored. If that value exceeds a given configurable threshold $th1$, then the queueing system starts discarding packets gradually. The packet drop probability gradually increases if the average queue sizes approaches a second threshold $th2$. If the average exceeds $th2$ then all packets are discarded. The assumption is that congestion is prevented thanks to the TCP congestion control mechanism which reduces the output rate in case of packet drop.

WRED (Weighted RED) is a more sophisticated dropping algorithm based on RED such that for each packet its drop probability depends on either the packet precedence or the DSCP value. For each type precedence or DSCP differentiation is achieved by specifying different values for parameters $th1$ and $th2$.

Core diffserv routers As explained in paragraph 5.1 for service 1, also in this case the diffserv network can be design in two different ways: by configuring scheduling in each output interface on the path between the server and clients or by the deployment of over-provisioning.
The distribution of diffserv functional blocks is similar to the one for Service 1 but in this case priority queueing is replaced by WFQ.

6 Conclusions and future work

Remote control applied to the high energy physics is a representative example of mission-critical research applications requiring the support of new and enhanced types of data transmission in order to be a reliable tool for researchers. In this paper we propose a quality of service architecture for the implementation of Quality of Service in packet networks by detailing first the application requirements and the corresponding service and by engineering the differentiated services network model in terms of placement and configuration of functional blocks like packet classification, marking, policing and scheduling. The network design here presented requires testing in a network testbed for the achievement of the ultimate goal: the deployment of advanced software tools in a production QoS-capable networking infrastructure.
The service implementation here described will be tested in a diffserv network scenario in order to validate and tune the QoS architecture here proposed.

References

[1] *The the Collider Detector at Fermilab:* `http://www-cdf.fnal.gov/`

[2] *The Rpc and Object Broker INterface* `http://www-b0.fnal.gov:8000/ROBIN/`

[3] *An Object Oriented Data Analysis Framework*, `http://root.cern.ch/`

[4] RFC 2475: *An Architecture for Differentiated Services*; S.Blake et al., Dec 1998.

[5] *A Conceptual Model for Diffserv Routers*; Y. Bernet, A. Smith, S. Blake, D. Grossman, March 2000

[6] RFC 2474: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*; K. Nichols et al., Dec 1998

[7] *A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks*; S.Jamin, P.B.Dunzig, S.J.Shenker, L.Zhang

[8] *An Admission Control Algorithm for Predictive Real-Time Service*; S.Jamin, S.Shenker, L.Zhang, D.D.Clark

[9] RFC 896: *Congestion Control in IP/TCP Internetworks*; John Nagle, Jan 1984

[10] RFC 2679: *A One-way Delay Metric for IPPM*; G. Almes, S. Kalidindi, M. Zekauskas, Sep 1999.

[11] *A Measurement-based Analysis of Expedited Forwarding PHB Mechanisms*; T. Ferrari, P. Chimento, IWQoS 2000, Pittsburgh June 2000, in print

[12] *Random Early Detection Gateways for Congestion Avoidance*; S.Floyd, V.Jacobson

[13] *RED behaviour with different packet sizes*; S. De Cnodder, O. Elloumi, K. Pauwels

[14] RFC 2598: *An Expedited Forwarding PHB*; V. Jacobson et al., June 1999.

[15] RFC 2597: *Assured Forwarding PHB Group*; J. Heinanen et al., June 1999.

[16] RFC 2698: *A Two Rate Three Color Marker*; J.Heinanen, R.Guerin, Sep 1999.

[17] *Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks*; H.Zhang.

[18] *The bandwidth guaranteed prioritized queuing and its implementations law*; K.L.E. Global Telecommunications Conference, 1997. GLOBECOM '97, IEEE Volume: 3 , 1997, Page(s) 1445-1449 vol3.

[19] *A comparative study of parallel and sequential priority queue algorithms*; R. Röngren and R. Ayani; ACM trans. Model. Comput. Simul. 7,2 (Apr. 1997), pages 157 - 209.

[20] *Priority Queuing Applied to Expedited Forwarding: a Measurement-Based Analysis*; T. Ferrari, G. Pau, C. Raffaelli, submitted to QofIS'2000, Mar 2000