Network Services Definition and Deployment in a Differentiated Services Architecture

E. Nikolouzou, S. Maniatis, P. Sampatakos, H. Tsetsekas, I. S. Venieris
National Technical University of Athens,
Department of Electrical and Computer Engineering,
9 Heroon Polytechniou str, 157 73, Athens, Greece
Telephone: +30 1 772 2551, FAX: +30 1 772 2534

Abstract- **Next generation Internet architectures will consider the Differentiated Services paradigm for the provision of quality of service to individual customers needs and applications. This paper addresses the definition and deployment of specific network services in a DiffServ environment. The proposed network services and the underlying traffic engineering methods are analyzed and simulated. Simulation outcomes prove that the fundamental principles of the network services are fulfilled.**

I. Introduction

The Internet is constantly evolving from a network carrying mainly data traffic into a network that should handle a variety of traffic profiles, ranging from real time audio and video to web traffic. However, the best effort nature of the current Internet is not sufficient to cope with the requirements of this traffic, in terms of throughput, delay, jitter and packet loss. The Integrated Services (IntServ) architecture [1] was the first significant step for the introduction of QoS in the Internet. IntServ uses the Resource Reservation Protocol (RSVP) [2] for the explicit setup of reservation state on each network node along the path from the sender to the receiver. However, the constant exchange of RSVP messages, as well as the need for separate reservation establishment for each flow raised scalability concerns. The Differentiated Services (DiffServ) architecture [3] emerged as a more scalable and manageable approach by providing relative prioritization of IP traffic. In DiffServ, IP flows with similar QoS requirements are grouped together under a common IP header field, the DiffServ Code Point (DSCP), and treated in the same queue inside the routers. DiffServ was also enhanced by the introduction of the Bandwidth Broker concept [4], a central entity that manages the resources of a domain and allocates them to requesting users.

The Differentiated Service architecture provides service differentiation based on the DSCP field in the IP header and the Per-Hop Behavior (PHB), which defines the externally observable behavior at each node. Two PHBs have been defined: the Expedited Forwarding (EF) [5] and the Assured Forwarding (AF) [6] PHB. The EF PHB provides premium service that can be viewed as a virtual leased line (VLL) service and is suitable for real time flows that require low delay, low jitter and guaranteed throughput. On the contrary, AF does not provide strict bandwidth guarantees, but assures that packets will be forwarded with higher priority in relation to best effort traffic. In the event of congestion, AF packets will encounter less bandwidth decrease than best effort traffic. There are four AF classed defined with three levels of drop precedence within each class, although it is not mandatory that all four classes have to be used in a domain.

In this paper we briefly present the Resource Control Layer (RCL), a distributed architecture for the efficient and manageable provision of QoS in DiffServ-based networks. However, we mainly focus on the definition of the Network Services that are offered to the users and their deployment on the network. In order to accommodate traffic with different QoS requirements, a limited set of Network Services has been defined. The proposed Network Services follow the concepts of the IETF EF and AF PHBs, but they additionally propose and exploit a specific implementation, which alleviates their deployment in real networks.

The paper is structured as follows. In Section II, the most important concepts of our architecture are described: the Resource Control Layer and the Network Services. Then, in Section III, the concept of Traffic Classes is introduced and described in detail. Traffic Classes provide the traffic handling mechanisms for each Network Service. Section IV presents the simulations performed and the obtained results that investigate the performance of the proposed Network Services in a realistic network topology.

II. Basic Architectural Mechanisms

A. The RCL Layer

The aim of this section is to give a short introduction of the architecture, in the context of which the Network Services are defined. The architecture consists of two functional areas: the data plane that is responsible for transmitting IP packets, and an overlay control plane, namely the Resource Control Layer (RCL) that is based on the Bandwidth Broker (BB) concept. Although the classical BB architecture proposes a concentrated approach where one BB is responsible for an administrative domain, RCL is designed as a distributed BB, to overcome scalability problems. As depicted in Fig. 1, the three key components of the RCL are: the Resource Control Agent (RCA), the Admission Control Agent (ACA) and the End-User Application Toolkit (EAT).

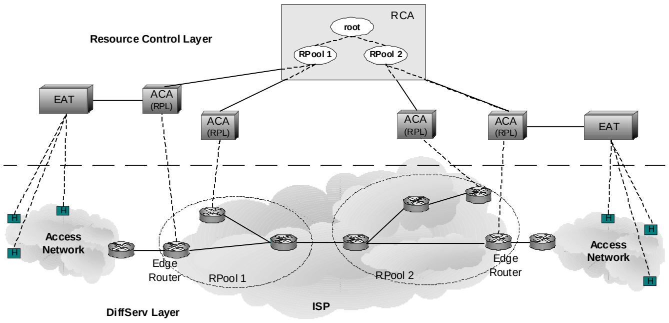

Fig.1: RCL Structure and main interactions

The Resource Control Agent is mainly responsible for the management of network resources (bandwidth). In order to simplify the task of the RCA to handle the resources efficiently, two concepts are introduced: the Resource Pools (RP) and the Resource Shares

(RS). The Resource Pool concept is based on the fact that in most networks the topology follows a tree like structure and therefore an aggregation of traffic on each subsequent level. According to the composite pattern [7], the RPs form a tree hierarchy as it is shown in Fig. 2. Each RP, which is a logical entity, is associated with a real sub-area of the network and manages the resources of the sub-area. The root of the tree is in charge of the available resources in the whole network, while each leaf of the tree structure (Resource Pool Leaf, RPL) is associated to one Admission Control Agent, which is in turn associated to one edge router (ER) of the network. In the provisioning phase, by taking into account the traffic forecasts, the complete network topology and the resource sharing policies between the network services, the initial values for the resource distribution are calculated. After the phase of provisioning each RP/RPL is initialized with a specific amount of resources that will allow making admission control decision locally. Within each RP/RPL, the total resources assigned to it are distributed among the available network services, in the form of Resource Shares. However, in many cases the initial distribution of resources differs significantly from the actual traffic load, thus an intelligent load-balancing redistribution algorithm has been defined to redistribute the available resources appropriately [8].

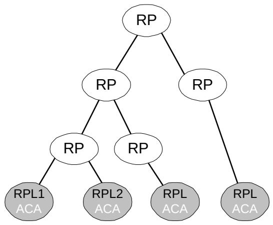

Fig. 2: Hierarchy of the RPs

In order to better understand the redistribution process and how it is related to the resource pool concept, consider the case where a RPL1 has already reserved all the resources that have been assigned to it initially, while RPL2 has reserved only a small amount. The "father" RP by identifying that situation will dynamically shift resources from RPL2 to RPL1. In other words, even if the provisioning phase produces less accurate values for the resource distribution, the proposed mechanism will identify the actual needs (based on the resource reservation requests) and adapt accordingly.

The Admission Control Agent is mainly responsible for user authentication and authorization, reservation handling, and flow Admission Control (AC). User authentication is initially performed through an explicit login procedure, while authorization refers to the permission of a user to place a request for the selected network service. During reservation handling, flow admission control is performed. The admission control decisions are made only at the edges of the network, therefore the corresponding ingress and egress points (ingress-egress ACAs) of the flow are identified and the Resource Share, which corresponds to the user selected network service, in the RPL is checked to ensure that the new flow can be accommodated. The core network is provisioned through the RP mechanism explained above, in order to ensure that once the admission control at the edges succeeds, no bottleneck will be created in the core network. Upon a successful reservation request, the corresponding ACAs consequently configure the edge routers appropriately to accommodate the new flow.

Reservation requests are forwarded to the ACAs from the End-User Application Toolkit, which mediates between end-users or applications and the network. The EAT interacts with the ACA to be aware of the available network services. A reservation request specifies the flow identifiers, the selected network service and the traffic profile for it. Special support is foreseen for legacy applications as well as for end users that are not aware of traffic description details, through the use of Application Profiles. Profiles are prepared through extensive testing of the application behavior and stored in a repository. The EAT interprets the profiles and prepares the reservation on behalf of the user.

B. Network Services

In order to provide QoS guarantees in a DiffServ network it is essential to assure QoS differentiation. Therefore, a set of five Network Services (NS) has been specified and implemented in our framework, which comprises the services sold by the provider to the potential customers, either end-users or other providers. They describe the QoS treatment a user's traffic experience within a network. The specified NSs are: Premium Constant Bit Rate (PCBR), Premium Variable Bit Rate (PVBR), Premium Multimedia (PMM), Premium Mission Critical (PMC) and Standard Best Effort (STD). Applications can be grouped into this relatively small number of services, with the applications in each service having similar requirements on the network in order to perform effectively and flows in each service having similar characteristics.

The PCBR network service is intended to support applications that require VLL-like services. Therefore, it is appropriate for voice flows, voice trunks or interactive multimedia applications. That kind of flows usually is characterized by low peak-to-mean ratio (almost constant bit rate, CBR) and low bandwidth requirements, while a great number of them are unresponsive (UDP). In addition, they should have small packets, so as not to provoke long transmission delays. It requires and expects to receive low delay, very low jitter and very low packet loss. The targeted quantitative value for end-to-end delay is less than 150msec for 99.99% of the packets, while packet loss is expected to be less than 10^{-6}.

The PVBR network service mainly copes with unresponsive variable bit rate (VBR) sources with medium to high bandwidth requirements. The intention is to separate those possibly high bandwidth VBR flows from the low bandwidth VBR and CBR flows in PCBR. This is caused by the fact that peak rate allocation is inefficient for the high bandwidth VBR flows, contrary to the flows belonging to PCBR. Typical candidate applications are real time video and teleconferencing. The requirements are similar to the PCBR network services but with a less strict need concerning the jitter and packet loss. They are characterized by large packet size and high peak-to-mean ratio. The targeted end-to-end delay is limited to less than 250msec for 99.99% of the packets, while packet loss should be less than 10^{-4}.

The PMM is expected to carry a mixture of TCP and non-TCP traffic. These flows require a minimum bandwidth, which must be delivered at a high probability. Independently of the transport protocol, flows are expected to implement some kind of congestion control mechanism and their aggressiveness should be similar to the one of TCP, assuming that they are roughly TCP-friendly [9]. This NS is supposed to serve adaptive applications (TCP), like low-quality video, streaming multimedia applications or file transfer (FTP). By nature, these flows are usually responsive, greedy and reflected to long-lived connections. They require throughput guarantees, which are translated into low packet loss for in-profile packets ($P_{loss} \leq 10^{-3}$), while there are no QoS guarantees for out-of-profile packets.

PMC is targeting to non-greedy adaptive applications that have great sensitivity concerning packet loss. It is thus suitable for transaction-oriented applications and interactive applications such as online games and chat-like applications. The main characteristics are the non-greediness of the flow, the responsive nature (TCP), the low use of bandwidth and the short life of the connection. As mentioned above, the most critical QoS parameter is the packet loss, so the most important requirement is very low packet loss for in-profile packets ($P_{loss} \leq 10^{-6}$), while no QoS guarantees are expected for out-of-profile packets. Nevertheless, low queuing delay is also desired, in order to retain the meaning of interactiveness.

Finally, packets belonging to the STD class receive no special treatment in the network.

III. Traffic Handling

The previous section introduced briefly the architecture and the proposed network services. This section covers some detailed network service aspects.

A. TCLS Implementation

The implementation of the Network Services is realized with the use of some network's mechanisms, which are the Traffic Classes (TCLs). A TCL is defined as a composition of a set of admission control rules, a set of traffic conditioning rules and a per-hop behavior (PHB). In the proposed architecture five TCLs are introduced: TCL1, TCL2, TCL3, TCL4 and TCL5 which correspond to PCBR, PVBR, PMM, PMC and BE. Each TCL maintains a separate queue at the router output ports and allocates one or more DSCPs in order to enable differentiation of packets in the core network. A PHB implemented in the output port of a router is realized in the network with the use of scheduling and buffer management algorithms. The scheduling mechanism selected is a combination of the Priority Queuing (PQ) [10] and Weighted-Fair Queuing (WFQ) [10], which is called PQWFQ and is depicted in Fig. 3. A weight is assigned to each TCL, though a queue is dedicated for TCL-1, which has strict priority over the other TCLs. The rest TCLs are scheduled with the WFQ and each queue is managed by different queuing strategy (Drop-Tail, Random Early Detection (RED), Weighted-RED (WRED) [11, 12]).

PQWFQ overcomes the limitations introduced by the PQ, which provides absolute preferential treatment to high priority traffic, while the lowest priority traffic (BE) is possible to experience starvation. On the other hand, WFQ would not be able to guarantee the strict delay requirement for TCL1 and TCL2.

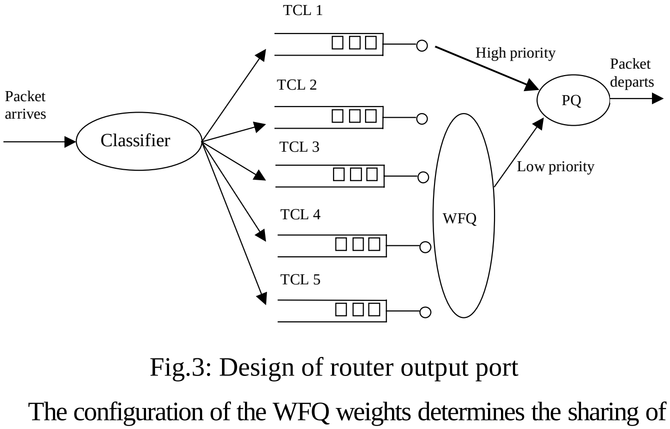

Fig.3: Design of router output port

The configuration of the WFQ weights determines the sharing of each link resources among the different traffic classes. In this way the network operator expresses its requirements on the network resources utilization and the maximum amount of traffic for each TCL, which is allowed to transit onto a link. The routers are configured with those weights during the start-up procedure. Those weights are somehow static, since are not updated dynamically. According to the WFQ weights, the AC rate limits for all TCLs at the RPLs are also set, which will be used by the AC algorithm.

Moreover, specific policing actions are deployed to ensure that non-conforming data flows do not affect the QoS requirements for already active data flows. Policing at the network access point is performed through a token bucket device (*r*,*b*) [13]. A specific traffic profile is determined for each NS, which best characterizes the data source.

The traffic profile for TCL1 is described in terms of a Single Token Bucket, which polices the peak rate of the flows. Admission control functions are also based on the peak rate of allocations for TCL1, since those flows are usually of low bandwidth. The single token bucket (TB) operates as both meter and dropper. Since TCL1 is characterized with strict QoS requirements, packet exceeding the declared profile should be dropped. The single token bucket is configured with token rate *r* equal to the *Peak Rate* (*PR*) of the flow, and bucket size *b* equal to a multiple *x* of the maximum allowed packet size (<256 Bytes), which is called *Bucket size for PR*, (*BSP*). The value of x lies in the range of {1,5}; a possible value could be x=1, while a larger value would allow a small amount of burstiness. The traffic conditioning mechanism is realized in the routers with the use of the Committed Access Rate (CAR) mechanism. Packets of TCL1 are enqueued in a single FIFO drop-tail queue.

Peak rate allocation is not appropriate for TCL2, since it is characterized with high bandwidth flows. Therefore, admission control function is based on both the peak and sustainable rate of the flows and a dual TB as meter and dropper is proposed. The first token bucket is configured with *r* equal to the *Sustained Rate* (*SR*) of the flow, and *b* equal to the *Bucket Size for SR* in bytes (*BSS*). The second token bucket is configured with *r* equal to the *PR of the flow* and *b* equal to a multiple *x* of the maximum allowed packet size (<1500 Bytes), (BSP). The value of x is in the range of {1,5}. The depth of the first bucket defines the burstiness allowed for the sender's flow (BSS). A packet is marked as in-profile if there are enough tokens in the first and second TB to accommodate it, otherwise it is dropped. The intention is to limit the sender's traffic in order to be conformant to the profile of the first TB (SR, BSS), while the second TB (PR) allows an amount of burstiness. Packets of TCL2 are also enqueued in a single FIFO drop-tail queue.

A single TB as a meter and marker is proposed as the traffic conditioning mechanism for TCL3, which polices the sustained rate and is configured with *r* equal to the *SR of the flow* and *b* equal to *BSS*. Flows conforming to this profile will be marked as in-profile otherwise as out-of-profile. The bucket size (BSS) should be very high to satisfy the bursty nature of TCP traffic and the maximum allowed packet size of flows could be set to 1500bytes. Packets of TCL3 are enqueued in a single FIFO queue, which is managed by WRED with two sets of parameters (minth, maxth, maxp). One set is for in-profile and the other for out-of-profile packets, as described in [12]. Out-of profile packets are not dropped, but marked with a different DSCP.

The traffic profile for TCL4 is specified with the use of a Dual Token Bucket, which polices both the sustainable and peak rate of flows. The first token bucket is configured with (*SR of the flow*, *BSS*), while the second token bucket with (*PR of the flow, BSP*). The parameter *x* for TCL4 has a fixed value in the range of {1,5} and the maximum allowed packet size can be set to 1500bytes. A packet that requires fewer tokens than available in the first and second TB is marked as in-profile, otherwise is marked as out-of-profile and forwarded into the network. SR should be small in order to disable greedy sources to transmit in-packets with a high rate into the network, while BSS should be large enough to allow several back-to-back packets to enter the network without being marked as out-of-profile. TCL4 occupies two DSCPs, one for in-profile and one for out-of-profile packets. WRED with two sets of parameters is used in order to discriminate out-of-profile packets against in-profile packets.

Finally the TCL5 requires no quality of service guarantees and best effort packets are enqueued in a single FIFO queue.

B. Admission Control Functions

Admission Control (AC) plays a significant role in ensuring the requested quality of service to user traffic. It is mainly responsible for limiting the access to the network, so that the already admitted flows do not anticipate any deterioration in their quality contract. Therefore, a bottleneck is prohibited to arise in the edge-link (i.e. the link between a core network and the ingress or egress router) as well as in any of the internal-links. The AC rate limits may be changed dynamically as a result of resource pool operations. Setting the AC limits is based on the target network utilization as well as the target performance of each Network Service. Although it is not the aim of this paper to analyze the AC procedures, the differentiation of AC functions among the proposed network services is briefly discussed.

Admission control is performed per flow and is based on the selected network service and the traffic profile from the user in the reservation request. According to the network service indicated in the request, the (ingress or/and egress) ACA module uses a specific formula in each case, as shown below:

$$PR_{new} + \sum_{1}^{N} PR_i \leq \rho C_{TCL1} \qquad \text{for TCL1}$$

$$Eff_{new} + \sum_{1}^{N} Eff_i \leq C_{TCL2} \qquad \text{for TCL2}$$

$$SR_{new} + \sum_{1}^{N} SR_i \leq C_{TCL3} \qquad \text{for TCL3}$$

$$Eff_{new} + \sum_{1}^{N} Eff_i \leq C_{TCL4} \qquad \text{for TCL4}$$

TCL1 is described in terms of a single TB, which polices the peak rate. The AC function therefore checks whether the PR of the new flow (PR_{new}) plus the sum of the PR of all admitted TCL1 flows do not overcome a threshold calculated by multiplying the capacity for TCL1 (C_{TCL1}) by a parameter ρ, which corresponds to the target utilization for TCL1. The capacity of a TCL is the Resource Share for this TCL in the RPL logically associated with the ACA.

TCL2 is based on a dual TB, so the peak and sustainable rates, along with the BSS are used for the calculation of the *effective bandwidth* for the new flow and the already admitted TCL2 flows. The sum of these must be less than the capacity of TCL2 in the RPL (C_{TCL2}), in order the new flow to be accepted. The formula for the calculation of the effective bandwidth takes into account the target packet loss rate and can be found in [14].

TCL3 is described in terms of a single TB, which polices the sustainable rate. The formula used in this case is similar to TCL1, but the SR is used in the calculations. Moreover, the parameter ρ is not used here. The fact that the AC algorithm is based on SR maximizes the network utilization. This is adequate for TCL3 as it provides only rough guarantees to the TCP-controlled flows that are supposed to be submitted in this traffic class.

A dual token bucket characterizes TCL4, so like TCL2, the effective bandwidth is calculated according to PR, SR and BSS, targeting at this time to a zero packet loss. The formula used for the calculation of TCL4 is different than that for TCL2 and can be found in [14].

IV. SIMULATIONS

This section presents some simulation results illustrating the quality of service offered by TCL1 and TCL2. The test topology is consisted of 5 routers connected in a chain for achieving a relatively large number of hops. The links are of high capacity (44Mbps), except from the access link, between the edge router, where the sources are connected, and the first core router, which compromises the bottleneck with a low capacity of 2Mbps. The Opnet 7.0 simulation tool is used for the simulations.

The topology is used for studying the performance of the different Network Services, under different traffic loads and under different scheduling algorithms; WFQ, PQ, PQWFQ and FIFO. The performance is based on measurements such as the one-way average delay and the packet loss.

A. Results for TCL1 – TCL2

Each link capacity is distributed among the traffic classes, which is determined by the assigned WFQ weights. For TCL1 a relatively small share of each link is recommended, which is restricted to 13% of the bottleneck link, determining that the maximum allowed traffic for TCL1 is 260Kbps. TCL2 reserves the 20% of the link capacity, while the rest (67%) is reserved by TCL5.

In the simulations, voice flows with bandwidth of 64Kbps are used for TCL1. Each packet size is 218bytes, including the relative protocol headers. Furthermore, the buffers for TCL1 in the routers were set to 10 packets to guarantee low packet delay requirements. Video flows for TCL2 use an exponential interarrival distribution with mean time 0.039sec, and constant packet size of 512bytes determining an average bandwidth of 105Kbps. Finally best effort flows for TCL5 are modeled with an exponential interarrival model, with mean time of 0.07sec and constant packet sizes of 1500bytes, where the average bandwidth of each flow is 171Kbps. The traffic load of TCL5 varies from a number of 4 admitted flows to a number of 11 admitted flows (684Kbps-1881Kbps), therefore utilizing its assigned bandwidth from 50% - 140%. When 8 BE flows are transmitted, then TCL5 is considered to occupy all of its reserved bandwidth, while when more than 8 flows are transmitted, then the bottleneck link is considered congested. When 11 flows of BE are transmitted, then TCL5 is considered to occupy 140% of its assigned bandwidth, therefore transmitting 540 Kbps additionally.

We assume a 74% utilization of the reserved bandwidth for TCL1, in order to guarantee its strict delay and packet loss requirements. For TCL2, 80% of the reserved bandwidth is utilized, providing TCL2 with its QoS requirements. The CAR is configured for both TCL1 and TCL2, as depicted in TABLE I, in order to police the admitted traffic, and drop packets in case they exceed the predefined profile.

TABLE I
CAR Profile Configuration

CAR profiles	
TCL1 (Single Token Bucket)	PR = 192Kbps, BSP = 218bytes

TCL2 (Dual Token Bucket)	PR=340Kbps, BSP=1024bytes,
	SR=315Kbps, BSS=5120bytes

The characteristics of the one-way average delay as a function of the packet size for TCL1 is given in Fig.4. In this case TCL2 is considered to occupy 80% of its reserved bandwidth, while four flows of the best effort traffic are transmitted. The maximum observed delay was 105msec, which is acceptable for voice traffic.

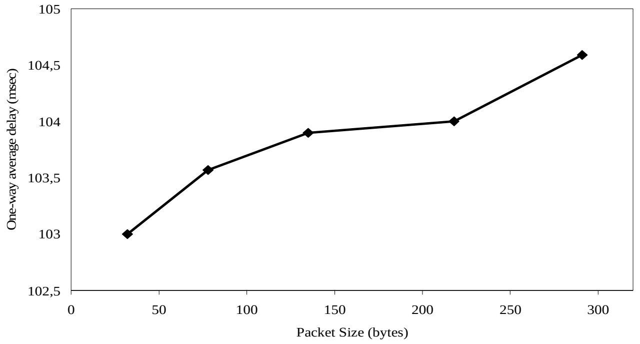

Fig. 4: One-way average delay of TCL1 vs. packet size

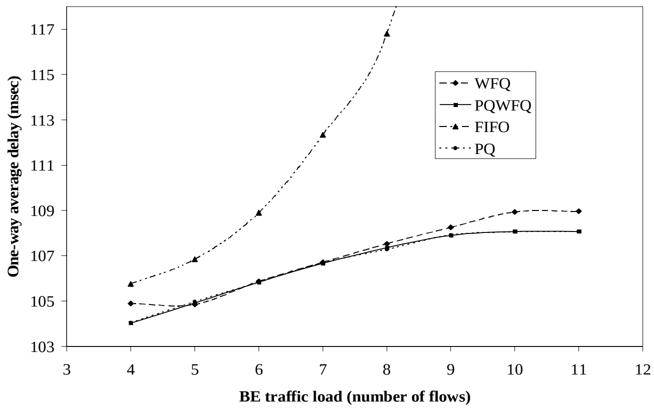

Fig.5: One-way average delay of TCL1 vs. different TCL5 (BE) traffic load and different scheduling algorithms

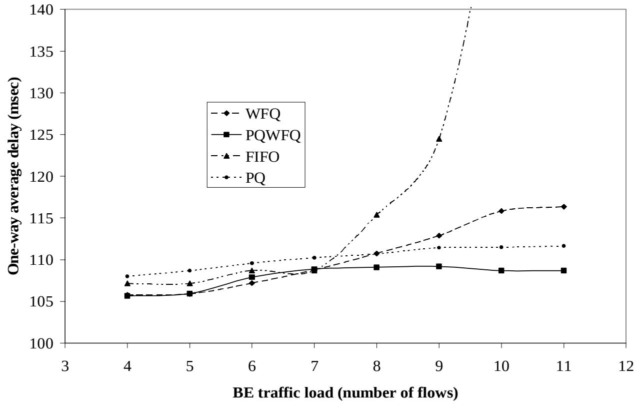

Fig. 6: One-way average delay of TCL2 vs. different TCL5 (BE) traffic load and different scheduling algorithms

The delay for TCL1 and TCL2 under different traffic load of TCL5 and under different scheduling algorithms is depicted in Fig. 5 and Fig. 6 respectively. In these figures, the delay using the FIFO scheduling algorithm gets a high value (214.4 msec for TCL1 and 231.9 msec for TCL2) when having 11 BE flows, and therefore due to lack of space is omitted.

PQWFQ guarantees for both voice and video traffic an accepted delay, which is quite lower than the maximum delay determined for each NS. Voice experiences a maximum delay of 108msec under a heavy load of TCL5, while it can tolerate up to 150msec. Video experiences a maximum delay of 109msec, which is an acceptable value comparing to the maximum value of 250msec. We have to stress here that also FIFO achieves a lower than the maximum value of end-to-end delay for TCL2 (231.9 msec), but this is accomplished with an unacceptable packet loss rate, as more than 20% of TCL2 packets are dropped under FIFO (Fig. 8).

TCL1 and TCL2 experience under PQWFQ better performance than under WFQ. This can be justified from the fact that in PQWFQ a queue is dedicated for TCL1, which has strict priority over the rest TCLs, while TCL2 achieves better utilization of its assigned bandwidth. PQ provides the best delay guarantee to TCL1, while degrades the performance of TCL2 and may drive TCL5 to starvation.

The packet loss for TCL1 and TCL2 is depicted in Fig.7 and Fig.8 respectively. The simulations assume the maximum allowed traffic for TCL1 and TCL2, and 11 flows of BE traffic (congested network). Under WFQ, all TCLs experience packet loss only when 11 flows of BE traffic are transmitted, while under PQWFQ and PQ only the best effort traffic experiences packets drops. FIFO provides the worst performance as depicted in Fig.7 and Fig.8.

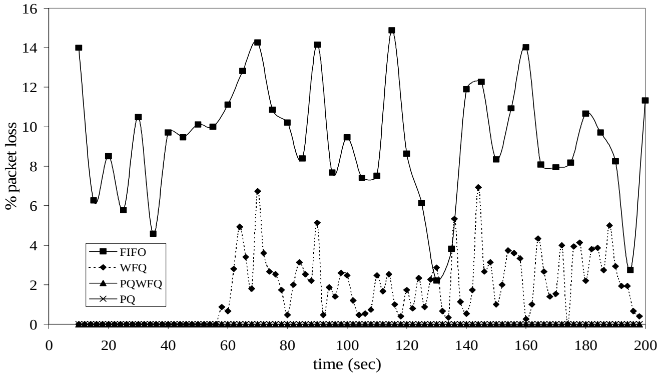

Fig. 7: Packet loss for TCL1 under 11 flows of BE

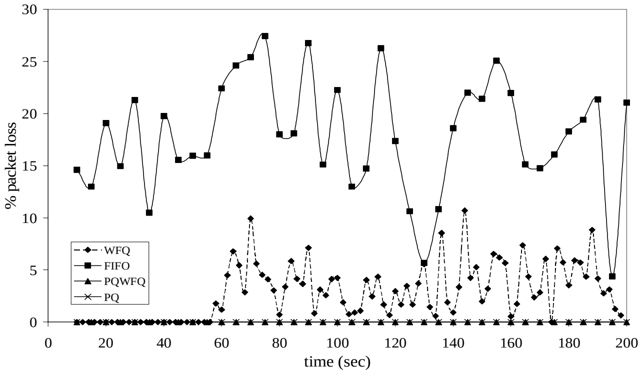

Fig. 8: Packet loss for TCL2 under 11 flows of BE

As far as the BE traffic is concerned, it experiences packet discards under all scheduling algorithms, while it receives the best performance under FIFO. This is caused by the fact that under FIFO all traffic flows experience the same behavior and no prioritization to any TCL is given. Therefore, the excess load of TCL5 traffic storms the bottleneck link, inducing greater packet discards to both TCL1 and TCL2.

B. Impact of TCL1 on TCL2

The impact of TCL1 traffic on TCL2 is illustrated on Fig. 9, showing the end-to-end delay for both TCL1 and TCL2 using the PQWFQ scheduling algorithm and having 8 BE flows.

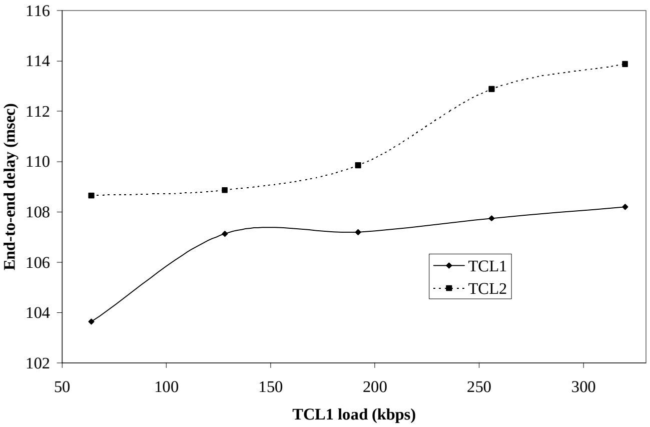

Fig. 9: End-to-end delay for TCL1&2 vs. different TCL1 load

It is observed that increasing the load of TCL1 above the admission limit (192Kbps) increases the end-to-end delay for TCL2, which reaches a value of 113,88msec. Likewise, the end-to-end delay of TCL1 is also increased, but in a small grade. This is due to the fact that for TCL1 a priority queue is dedicated providing to it strict priority over the other TCLs. Nevertheless, the end-to-end delay of TCL2 does not increase dramatically, since it utilizes its unused assigned bandwidth.

V. Conclusions

The work presented in this paper dealt with the definition and deployment of a set of Network Services within a DiffServ-enabled core network architecture. The Network Services, which are implemented in the network with the traffic handling mechanisms offered by respective Traffic Classes, target at different kinds of user traffic that exhibit similar QoS requirements and characteristics, and they therefore demand analogous treatment within the network.

We propose five Network Services that can accommodate most of the well-known application traffic usually submitted in a network. We described a specific implementation for all Network Services in the context of the Resource Control Layer architecture. Subsequently, simulation results proved that the proposed traffic handling mechanisms are adequate for two of the Network Services, the PCBR and PVBR. Future work is intended for the other two Network Services (PMM and PMC), in order to show that the main requirements for these services are also fulfilled.

References

[1] R. Braden, D. Clark and S. Shenker, “Integrated Services in the Internet Architecture: an Overview”, RFC 1633

[2] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, “Resource ReSerVation Protocol (RSVP)”, RFC 2205

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, “An Architecture for Differentiated Services”, RFC 2475

[4] K. Nichols, V. Jacobson and L. Zhang, “A Two-bit Differentiated Services Architecture for the Internet”, RFC 2638

[5] V. Jacobson, K. Nichols and K.Poduri, “An Expedited Forwarding PHB”, RFC 2598

[6] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597

[7] E. Gamma, R. Helm, R. Johnson, J. Vlissidis, *Design Patterns – Elements of Reusable Object-Oriented Software*, pp. 163-173, Addison-Wesley, 1995.

[8] E. Nikolouzou, G. Politis, P. Sampatakos, I. Venieris, “An adaptive algorithm for resource management in a differentiated services network,” *ICC2001*, Helsinki, Finland, June 2001.

[9] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control”, RFC 2581

[10] M. Markaki, E. Nikolouzou, I. Venieris, "Performance Evaluation of Scheduling Algorithms for the Internet", 8th IFIP Conference on Performance Modeling and Evaluation of ATM & IP Networks, Ilkley, June 2000

[11] V. Firoiu, M. Borden, “A study of active queue management congestion control,” *Infocom 2000*, Tel-Aviv, Israel, March 2000.

[12] T. Ziegler, C. Brandauer, S. Fdida, "A quantitative model for parameter setting of RED with TCP traffic", 9th International Workshop on Quality of Service, Karlsruhe, Germany, June 6-8, 2001

[13] Y. Bernet, S. Blake, D. Grossman,A. Smith, “An informal management model for Diffserv routers”, draft-ietf-diffserv-model-06.txt, February 2001, exp. August 2001

[14] Deliverable D1301, Specification of traffic handling for the first trial, AQUILA project consortium, http://www-st.inf.tu-dresden.de/aquila/, September 2000.