Service Level Management of Differentiated Services Networks with Active Policies

Paulo Rogério Pereira[1], Djamel F. H. Sadok[2], Paulo Pinto[3]

[1]Instituto Superior Técnico, Universidade Técnica de Lisboa. INESC ID, Rua Alves Redol, 9. 1000-029 Lisboa, Portugal.
Phone: +351-213100345. Fax: +351-213145843. Email: prbp@inesc.pt

[2]Universidade Federal de Pernambuco. Centro de Informática. Caixa Postal 7851. CEP 50732–970. Recife, PE, Brazil.
Phone: +55-81-32718430. Fax: +55-81-32718438. Email: jamel@cin.ufpe.br

[3]Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. UNINOVA, Quinta da Torre.
2825 Monte de Caparica, Portugal. Phone: +351-212948300. Fax: +351-212948532. Email: pfp@uninova.pt

Abstract

This paper describes a hierarchical architecture of active policies that performs the management of a differentiated services network to provide the best possible quality of service to users and help fulfilling service level agreements with minimum costs to the service provider.

I. INTRODUCTION

The highly competitive field of providing telecommunication and network services to users calls for the use of advanced network and systems management techniques. These techniques should dynamically optimize every aspect of the equipment and services involved to provide the best possible quality of service to users. This quality might even be formalized through a Service Level Agreement (SLA) between provider and users.

This paper describes a hierarchical architecture of active policies that performs the management of a differentiated services (DiffServ) [1] network to attain the aforementioned objectives.

Active policies are active objects with internal state and a set of operations, specialized on managing a certain problem over a subpart of the network [2]. The conditions they take care of are formalized on a *contract* to the entity that launched the policy. While working, the active policies make the current status of their activity available through *management quality of service* parameters that provide an indication of how well they are performing their task.

The results presented in this paper were obtained with VINT project's *network simulator* [3], with a modified version of DiffServ patches [4], for the 15-node network of [7]. The results show that the use of active policies improves the service offered to users, by constantly adapting to the network state, and helping to fulfill service level agreements with minimum costs to the service provider.

The next section describes the system architecture with a brief description of the policies. The following section provides details about the most important active policies, and the corresponding simulation results. The final section draws some conclusions and raises further work topics.

II. SYSTEM ARCHITECTURE

Figure 1 shows the active policies deployed and how they are related to each other. The different levels correspond to different abstraction levels, allowing a refinement of the quality requirements.

At the Equipment abstraction level, the traffic class weight management policy dynamically adjusts the relative weights of each of the DiffServ traffic classes on each device individually. This policy's contract is that the quality of service (QoS) (delay, jitter, throughput) is the best for the Expedited Forwarding (EF) class [5], followed by the higher priority Assured Forwarding (AF) class [6], and the worst for the Best Effort (BE) class. This policy also manages out-of-profile AF packet discard to help prevent line overload. Presently, only one AF class, AF1, is being used.

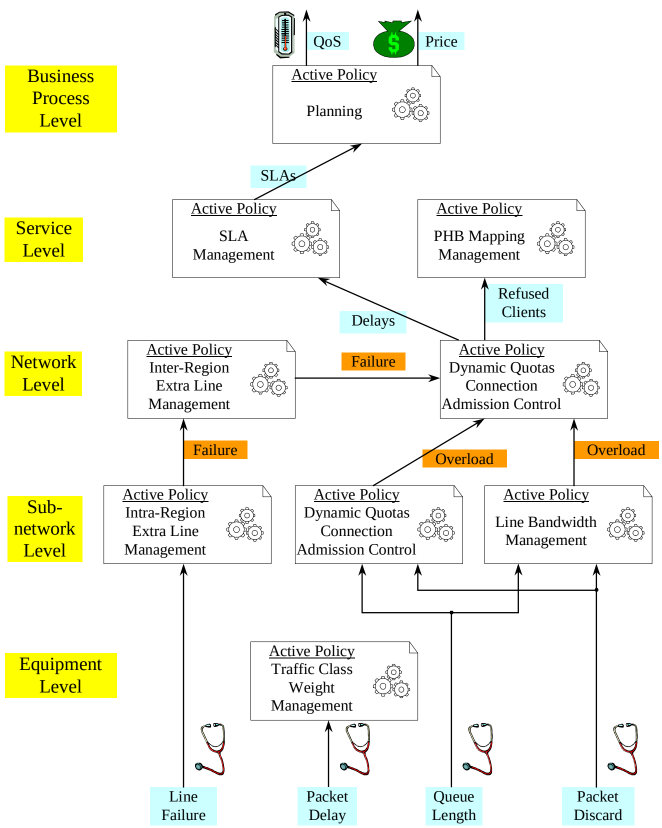

Fig. 1 Active Policies used

The Network level is sub-divided in two sub-levels to ease implementation and improve scalability. The Sub-network level policies are responsible for a certain region, and the network level policies act over the set of all the regions. The network is divided in several regions to improve geographical independence and reduce management traffic.

The contract at the Network level is to maintain connectivity between all nodes with no packet loss on the EF class, and very low packet loss for inside-profile AF packets.

The extra line management policies presented in [7] are responsible for managing the activation of backup lines whenever there is a line failure or line overload that causes a local QoS degradation. The sub-network level version deals with problems within a region, and the network level version deals with problems that affect more than one region.

The line bandwidth management policy dynamically adjusts each network core line bandwidth to maintain the required QoS for each individual line. It assumes that the service provider has high capacity lines in the network core that are shared between voice and several data networks, so that the bandwidth reserved for the DiffServ network can be modified as needed within certain limits.

The admission control policies evaluate the network load to determine where and when further connections can be allowed into the AF traffic class without degrading the QoS that should be provided by the network, even when the bandwidth can not be further increased. The EF traffic class uses a bandwidth broker for admission control, and the BE class always accepts all connections.

Going up one level, at the Service level, the contract is to ensure the negotiated throughput and average delays in the EF and AF classes, and to maximize service availability reducing connection refusals.

The PHB mapping management policy controls the local mapping of the different user application traffic types into the DiffServ traffic classes (or per-hop-behavior, PHB). This mapping can be changed to reduce connection admission refusals, or reduce load in overload situations.

The SLA management policy monitors the end-to-end QoS being obtained by the users. If the QoS is not satisfactory, even in spite of the lower level efforts, this policy adjusts lower level policy parameters to optimize performance and enforce the SLAs.

Finally, the planning policy at the Business Process level performs long term adjustments in lower level policy parameters to improve SLA enforcement.

III. ACTIVE POLICIES

A. Line Bandwidth Management

The line bandwidth management active policies are local to each core line. Each policy calculates an internal management QoS parameter to account for line load as follows:

$$QoS = EFpacketDiscard + AFinProfilePacketDiscard + \\ + EFqueueLength * 3 / EFqueueSize + \\ + AFqueueLength * 4 / AFqueueSize \quad (1)$$

The QoS value is larger as the queue length increases, or packets that should not be lost are discarded. The AF queue uses a random early detection (RED) mechanism with in/out profile bit [8]. The RED thresholds are configured so that the QoS value due to the AF queue size is on average 0.5 when out-of-profile packets start being discarded, 1.0 when all out-of-profile AF packets are discarded and in-profile packets start to be discarded, 2.0 when all AF packets are discarded. The use of the instantaneous queue length instead of the average queue length ensures a faster response.

The line bandwidth is adjusted between 2 Mbps and 30 Mbps according to figure 2 thresholds at 25 ms intervals. If the QoS is below 0.4, the bandwidth is decreased by 0.5 Mbps. If the QoS is between 1.0 and 1.5, the bandwidth is increased by 1 Mbps. If the QoS is between 1.5 and 2.5, the bandwidth is doubled, and if the QoS is above 2.5, the bandwidth is set to 30 Mbps. If it is not possible to further increase the bandwidth, an overload notification is generated. The network uses shortest path first routing. If QoS routing was used, the update interval could be increased.

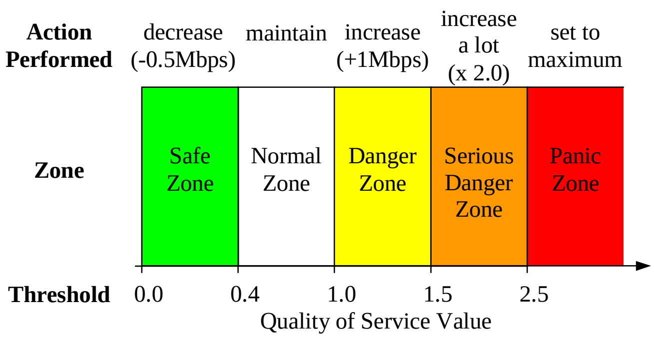

Fig. 2 Line bandwidth management policy operation

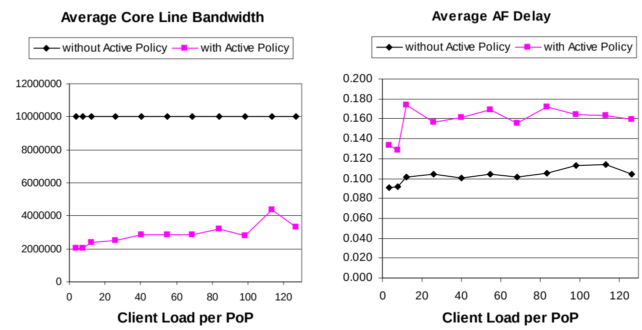

Fig. 3 Average core bandwidth and AF Delay

Figure 3 shows the average bandwidth for the core lines as a result of using this active policy for different network loads. This situation is compared with the case where the line bandwidth is fixed at 10 Mbps. For 10 Mbps lines, the network is fully loaded with about 15 connections per access node, or point of presence (PoP). The network was tested with much larger loads to analyze its overload behavior. The active policy continuously adapts the bandwidth so that the average bandwidth is much lower than in the constant bandwidth case, resulting in lower costs for the service provider.

Whether this active policy is used or not, a bandwidth broker is used for access control for the EF and AF classes. The simulations showed no packet loss for EF, or in-profile AF traffic. The disadvantage of decreasing the bandwidth with this active policy is that packet delay increases as shown in figure 3 for the AF case, which also causes smaller out-of-profile AF throughput.

B. Dynamic Quotas Connection Admission Control

The dynamic quotas connection admission control active policies are local to each PoP where users connect, replacing the bandwidth broker for the AF class. Each policy calculates an internal management QoS parameter to account for line load as follows:

$$QoS = AFinProfilePacketDiscard + 4 * \max(averageAFincomingQueueLeng, averageAFoutgoingQueueLeng) / AFqueueSize \quad (2)$$

The QoS value is larger as queue length increases, or packets that should not be lost are discarded. The maximum number of connections admitted is adjusted according to the thresholds of figure 4 at 1 second intervals, with a minimum equal to the value used for the bandwidth broker case. If the QoS is below 0.25, one further connection may be admitted. If it is between 0.25 and 0.75, no further connections are admitted. If the QoS is larger than 0.75, when a connection finishes, it is not replaced by a new one.

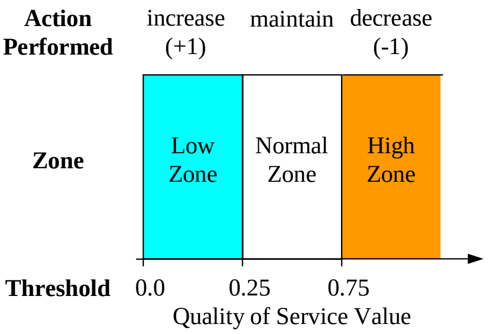

Fig. 4 Connection admission control policy operation

Fig. 5 AF class average traffic and throughput

The connection admission control active policy at the network level only acts by temporarily suspending new AF connection admissions in overload situations, propagating this command to the sub-network level connection admission control active policies.

Figure 5 shows the average number of clients and the average throughput for the AF class with and without the use of this active policy. As can be seen from these graphics, the bandwidth broker limits the connections admitted for the AF class to 4.1 on average per PoP, while the active policy allows more connections as load increases. For this simulation every user requested a 64 Kbps connection. The throughput graphic shows that this rate is guaranteed in both cases, but less out-of-profile traffic goes through, as load increases, especially in the active policy case.

For the service provider, this active policy saves money as more connections are admitted and the requested rate is still assured.

C. PHB Mapping Management

Traffic is mapped into the available traffic classes according to rules configured into packet classifiers. If the required class is full, the connection is established in the next available class. The PHB mapping management active policies are local to each PoP, selecting the most important connections that can use the higher priority traffic classes, and downgrading the least important ones, according to the network load.

Table 1 shows four possible restriction rules. Rule 1 makes no restriction. Rule 2 forces FTP traffic to use the BE class. Rule 3 additionally stops HTTP traffic from using the EF class. Finally rule 4 forces FTP and HTTP traffic to use the BE class, and only allows CBR traffic to use the EF class. Each rule further restricts the types of traffic that can use the higher priority traffic classes.

Table 1 PHB mapping restriction rules

Application	Rule 1	Rule 2	Rule 3	Rule 4
Telnet	EF AF BE	EF AF BE	EF AF BE	AF BE
CBR	EF AF BE	EF AF BE	EF AF BE	EF AF BE
OnOff	EF AF BE	EF AF BE	EF AF BE	AF BE
HTTP	EF AF BE	EF AF BE	AF BE	BE
FTP	EF AF BE	BE	BE	BE

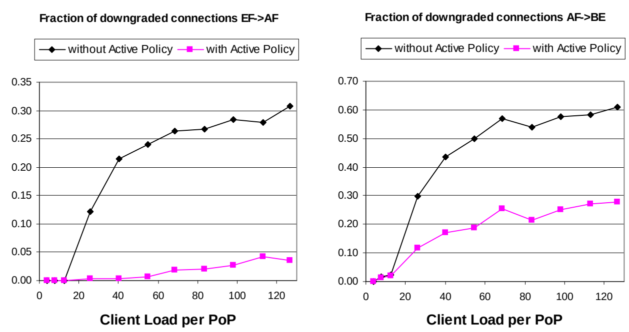

Fig. 6 Fraction of downgraded connections

The PHB mapping management active policies monitor the downgraded EF and AF connections. They change the rule in the direction 1 → 2 → 3 → 4 each time a connection has the requested class downgraded. Additionally, they change the rule in the direction 4 → 3 → 2 → 1 each time ten connections do not have the requested class downgraded.

Figure 6 shows the fraction of downgraded connections with and without this active policy for the EF and AF traffic for increased client load. These graphics show that this policy can significantly reduce the number of downgraded connections for the connections that were selected to use the higher priority traffic classes. This is achieved by selecting a different mapping that keeps only the most important connections in the higher priority traffic classes.

D. SLA Management

The SLA management active policy works over the global network to help fulfill SLA agreements. This policy monitors the end to end delays for the EF and AF traffic classes every second, acting when these delays approach the SLA, by adjusting lower level policy parameters:

- In the traffic class weight management active policy, lowering out-of-profile AF RED thresholds. This increases out-of-profile packet drop.
- In the line bandwidth management active policy, increasing line bandwidth.
- In the extra line management active policy, activating extra lines.
- In the connection admission control active policy, decreasing the maximum number of admitted connections for the AF class.

Figure 7 shows the average delays and core bandwidth obtained with an *SLA*(*delayEF*) of 60 ms, an *SLA*(*delayAF*) of 120 ms, and all policies up to the service level operating. These results show it is possible to enforce the SLAs with some error, still using a reduced bandwidth, which means reduced costs to the service provider. The BE class gets the remaining network resources.

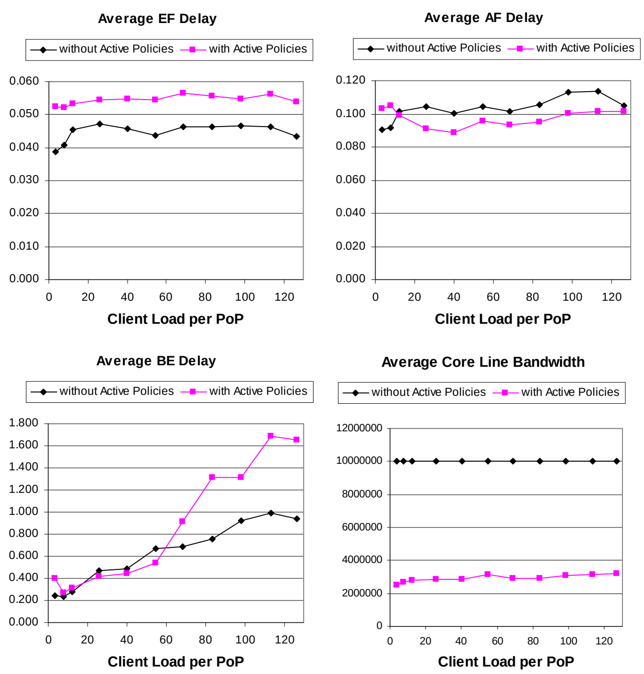

Fig. 7 Average delay and core bandwidth

Figure 8 shows the average traffic carried in the EF and AF traffic classes for increased loads. For light loads (less than 15 clients per PoP), the network is not fully loaded, causing both situations to look similar. For larger loads, the active policies lines in the graphics show the simultaneous effect of the PHB mapping active policies that reduce the load offered to both classes, and the connection admission control active policies that allow additional connections to be accepted for the AF class.

It can be concluded that the combination of all the active policies allows offering an improved quality of service to users by fulfilling the different abstraction levels' contracts.

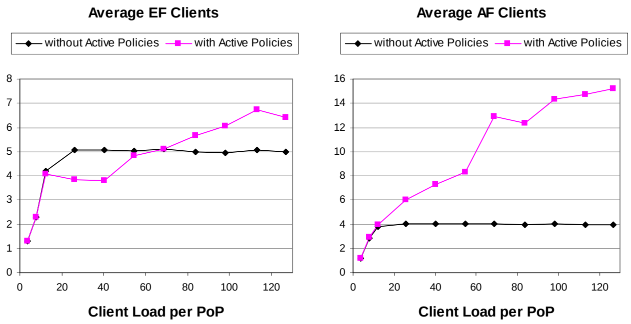

Fig. 8 Average EF and AF traffic

These simulations were repeated for three additional pairs of *SLA*(*delayEF*) and *SLA*(*delayAF*), as shown in table 2. The maximum average delay, regardless of the load, for each case is a measure of the QoS being provided to users. The average bandwidth is a measure of the cost to the service provider.

These results show that it is possible to enforce different SLAs within certain ranges. The first case shows average delays that are 25% of the SLA values. The next two cases have about 10% error. For the last case, the average delays are above the SLA, due to the importance of propagation delays. In general terms, as the maximum delays imposed are lowered, the bandwidth needed naturally increases. This increase is very substantial if the SLA can not be met, as in the last case. Note that the value exceeds 30 Mbps because extra lines were used and a correction factor was accounted for.

Table 2 Bandwidth and delays with the SLA policy

SLA (delayEF)	*SLA (delayAF)*	Max *delayEF*	Max *delayAF*	Average Bandwidth
200 ms	800 ms	75.4 ms	209.0 ms	3.186 Mbps
60 ms	120 ms	56.4 ms	104.9 ms	2.921 Mbps
40 ms	70 ms	45.5 ms	64.4 ms	6.738 Mbps
25 ms	30 ms	38.9 ms	39.6 ms	35.537 Mbps

E. Planning

The planning active policy adjusts the SLAs of the SLA management active policy to obtain better long-term results.

Both *SLA*(*delayEF*) and *SLA*(*delayAF*) are modified by an integral feedback system given by the equation:

$$SLA(n) = SLA(n\text{-}1) + K\,[SLArequired - delay(n)] \qquad (3)$$

where n is the sampling interval, K is the integrator gain, set to 0.25, to make slight changes in the SLAs for each sampling period. A small sampling period of 5 seconds was selected to ensure short simulations. In a real network, larger values should be used, more consistent with the SLA verification period.

An anti-windup mechanism was added to limit the SLA values to a range of 0.25 to 4 times the required SLA.

Figure 9 shows the average delays obtained with this policy as compared with those of the previous subsection. These graphics show that the static error is greatly reduced by the integrator, so that the EF delay is much nearer the required value of 60 ms, and the AF delay of 120 ms.

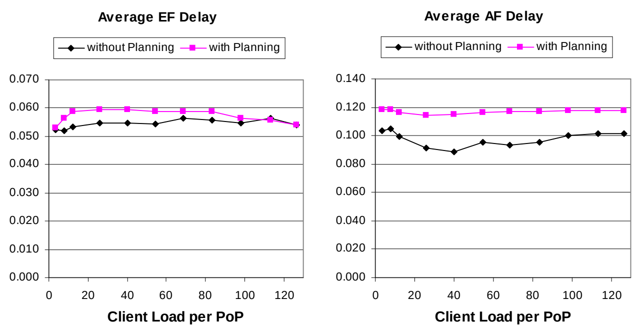

Fig. 9 Average EF and AF delay

The simulations of table 2 were repeated with the planning active policy operating. The results are presented in table 3, showing an improvement in SLA fulfillment. The two middle cases show about 1% error, instead of 10% for the previous situation. Naturally, a reduction of the delays is obtained at the cost of an increased average bandwidth, which means more costs to the service provider. However, in the first case of table 3, the bandwidth actually increased. This is due to more connections being accepted, as less restrictions are imposed, which results in larger bandwidth needs. In this case the costs of the bandwidth are covered by the increased profit due to the larger number of connections.

Table 3 Bandwidth and delays with the planning policy

SLA (delayEF)	*SLA (delayAF)*	Max *delayEF*	Max *delayAF*	Average Bandwidth
200 ms	800 ms	75.6 ms	210.3 ms	3.423 Mbps
60 ms	120 ms	59.5 ms	118.3 ms	2.902 Mbps
40 ms	70 ms	40.5 ms	69.4 ms	8.843 Mbps
25 ms	30 ms	35.8 ms	36.4 ms	38.988 Mbps

IV. CONCLUSION

From the results presented, it can be concluded that the quality of service required by users can be provided within certain limits by using active policies.

The active policies are useful in dynamically adjusting network parameters to optimize network performance and enforce service level agreement with minimum costs to the service provider.

Some of the active policies can be reused for different situations just by changing the variables they control and the configuration parameters, like the thresholds in the previous examples.

The works of [9][10] also present policy hierarchies. However, these works do not integrate with service level management, nor refine the requirements through several policy abstraction levels that act over the network down to the equipment, nor allow easy policy reuse. The policies of [11] restrict the network behavior, instead of improving its operation as in the case of the active policies.

Further research topics include defining a contract specification language, an active policy definition language, and applying the active policies to other examples.

V. REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[2] Paulo Pereira, Paulo Pinto, "Algorithms and Contracts for Network and Systems Management", in *IEEE Latin American Network Operations and Management Symposium*, Rio de Janeiro, Brazil, 3-5 December 1999, pp. 385-396. ISBN: 85-900382-3-8. http://mariel.inesc.pt/~prbp/publica/lanoms99.pdf

[3] UCB/LBNL/VINT Network Simulator (version 2). http://www.isi.edu/nsnam/ns/

[4] DiffServ additions to NS. http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/index.html

[5] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", IETF RFC 2598, June 1999.

[6] J. Heinamen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", IETF RFC 2597, June 1999.

[7] Elionildo Menezes, Djamel Sadok, Judith Kelner, Paulo Pereira, Paulo Pinto, "Service Management for Differentiated Services Networks", in *IEEE Workshop on IP-Oriented Operations & Management*, Cracow, Poland, 4-6 September 2000, pp. 99-108. ISBN: 83-88309-00-5. http://mariel.inesc.pt/~prbp/publica/ipom2000.pdf

[8] Paulo Rogério Pereira, Bruno Afonso, Daniel Gomes, "Differentiated Services Network Simulation", in *ConfTele'2001 proceedings*, Figueira da Foz, Portugal, April 2001.

[9] René Wies, "Policies in Network and Systems Management - Formal Definition and Architecture", *Journal of Network and Systems Management*, Plenum Publishing Corp., **2**(1):63-83, March 1994.

[10] Thomas Koch, Christoph Krell, Bernd Krämer, "Policy Definition Language for Automated Management of Distributed Systems", in *2nd International Workshop on Systems Management*, IEEE Computer Society, 19-21 June 1996, Canada.

[11] M. Sloman, J. Magee, K. Twidle, J. Kramer, "An Architecture for Managing Distributed Systems", in *4th IEEE Workshop on Future Trends of Distributed Computing Systems*, Lisbon, Portugal, IEEE Computer Society Press, pp. 40-46, 22-24 September 1993.