A Simulation Study of Web Traffic over DiffServ Networks

D. Rossi, C. Casetti, M. Mellia
Dipartimento di Elettronica
Politecnico di Torino, Torino, Italy
E-mail: {rossi, casetti, mellia}@mail.tlc.polito.it

Abstract—**In this paper we present a simulation study of HTTP traffic crossing a DiffServ domain. We consider both the cases where the reserved bandwidth is not exceeded by the offered traffic (overprovisioning) and where the assured traffic competes with the classic Best Effort class (underprovisioning). Simulation reported shows that DiffServ approach is able to protect the assured flows in the first case, while the performance benefits are tighter in the second case, in which fairness issues arise between long and short-lived flows.**

I. Introduction and Background

In recent years, researchers and service providers alike have looked at ways to overcome the shortcomings of the Internet Best Effort service. Two approaches have been widely touted as the solution to QoS requirements: IntServ [1] and DiffServ (DS) [2]. On the one hand, scalability issues have so far questioned the effectiveness of IntServ, which requires per-flow signaling and per-flow guarantees. On the other hand, DiffServ handles flow aggregates, performing packet classification into classes at the network ingress and supporting different per-class guarantees at every hop in the network core. Traditionally, DiffServ comes in two flavors: *Expedited Forwarding* (EF) [3], which is also called 'Virtual Wire', providing an almost airtight separation between premium and non-premium traffic; *Assured Forwarding* (AF) [4], in which different classes are given different forwarding and dropping treatments, although they share the same network resources. The excellent scalability properties offered by DiffServ have promoted its use among ISPs; however, it still remains to be seen whether DiffServ offers a sufficiently high degree of protection to privileged classes of traffic. In particular, the use of Active Queue Management (AQM) techniques in routers is a potential liability, what with the difficult choice of parameters that AQM usually entails.

While many studies [5], [6], [7], [8] have shown the benefits and drawbacks of the DiffServ scheme for long-lived TCP traffic, in this paper, we analyze a few plausible scenarios mixing AF and Best-Effort (BE) for short-lived flows, such as web-like traffic, and study the behavior of a DiffServ network implementing Assured Forwarding, in both underload and overload conditions; specifically, we have focused on the case when the AF traffic exceeds the amount of bandwidth specified (and guaranteed) in its contract.

This work was supported by a contract with the NEC Network Laboratories of Heidelberg, Germany.

The paper is organized as follows: Section II outlines the model used to simulated Web connections and the network topology and scenario; the numerical settings chosen in our study and the metrics of interest are included in Section III; simulation results for the cases under investigation are shown and discussed in Section IV. Section V concludes the paper, outlining what we believe are the pros and cons of the DiffServ model.

II. Simulation Scenario and Traffic Model

The simulations performed in this study were run under *ns*-2.1b8a [9] using the network configuration sketched in Fig. 1, which represents an ISP network collecting traffic from two regions: sources in the DS region (or "cloud") send DS-marked traffic, while sources in the BE region generate unmarked traffic. Routers S and R connect senders to receivers, and have identical configurations: i.e., a TSWTCM marker [10] as traffic conditioner, followed by a RIO-C queue [11] feeding a round-robin scheduler. TSWTCM marker parameters used for all the simulation presented in this paper identify two distinct classes of service, namely red or green; RIO-AQM parameters correspond to a staggered set, and for green and red packets $min_{th}/max_{th}/max_p$ are set to $20/40/0.02$ and $5/10/0.20$, respectively.

Each cloud produces unidirectional RENO-TCP flows (whose offered load is determined by both the overall normalized load ρ, and the per-cloud load fractioning α_i); ACKs are never lost on the backward path.

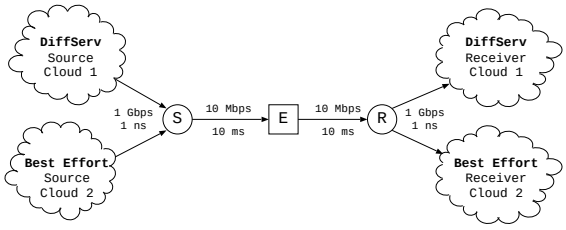


Fig. 1. Network Scenario

The traffic patterns used in the simulation are primarily short-lived flows – modeling web-like traffic, while endlessly-sending FTP sources can also be present as background traffic. The Web traffic generator activates

TCP flows trying to transfer an amount of data, λ, uniformly chosen among a coarse set $\underline{\lambda}$ of empirically selected flow lengths [5], [12]. The flow interarrival time is determined by i.i.d. exponential random variables with average τ_i, as summarized below:

$$\begin{cases} B = 10\ Mbps & \text{Bottleneck Bandwidth} \\ \rho & \text{Total Offered Load} \\ \alpha_i & \text{Per-Cloud Load, } \sum_k \alpha_k = 1 \\ \overline{\lambda} = 13.6\ KB & \text{Average Flow Length} \\ \underline{\lambda} & \text{Empirical Flow Length Vector} \\ \overline{\tau_i} = \overline{\lambda}/(\alpha_i \cdot \rho B) & \text{Average Interarrival Time} \end{cases}$$

This model is based on realistic traffic pattern and provides a simple means to investigate the impact of Diff-Serv on short-lived bursty traffic.

III. Experimental Design and Performance Metrics

Although the traffic sources are heterogenous, here we will focus only on HTTP-emulated connections. Beside the BE service, the DS domain offers an unique AF service, in which the guaranteed network resource is a bandwidth measure: the service is thus entirely specified in terms of a Committed Information Rate (CIR). Here, we will focus on the case –further indicated with SLA_{50}– where half of the bottleneck bandwidth is sold by the ISP to the AF flows and the excess is available to the sharing among AF and BE flows[1].

Simulation of bursty, short-lived flows need long runs in order to be confident with the obtained results. In our study, each simulation is run for at least $t_{stop} = 600s$, in the sense that no new flows are scheduled after t_{stop}; the simulation ends either when the last flow has completed its transmission or, forcefully, at $t_{halt} = 900s$, regardless of the presence of ongoing flows. Usually, all flows ended before t_{halt}. In this paper we investigate and compare DS and BE performance in two significant scenarios:

- **Overprovisioning of HTTP Traffic**

In this case, the total offered load ρ of web traffic varies, whereas α_{AF} is fixed and DS-HTTP traffic is substantially in-profile. We also examine the effects introduced by the presence of FTP background sources, which, being greedy connections, will push the bottleneck link always in congestion.

- **Underprovisioning**

In this case, the total offered load is fixed, while the percentage of AF traffic α_{AF} varies. This allows us to examine the performance of AF traffic when it exceeds its committed traffic profile.

In order to evaluate the ability to provide QoS to Web traffic, we choose as primary user-centric metric to monitor the flows completion time, i.e., the time required to completely receive all the flow packets; operator-centric metrics such as packet drop, TCP dynamics (cwnd, fast recovery, retransmission timeout) and RED-AQM early drop (the percentage of time a packet is discarded by the random mechanism instead of buffer overflow) complete the performance analysis.

All the performance metrics have been collected on a per-flow basis and all the statistics are averaged over at least 20,000 samples. We report results for the *Completion Time*, the *Packet Drop Percentage* and *Early Drop Fraction* versus the offered load. Results are reported for AF traffic in the bottom pictures, while the corresponding performance figure for BE traffic is plotted in the top picture. All figures plot the results for three different flow length classes: being π the number of packets to be transmitted by a flow, we report the averaged results for $\pi = 1, 10, 90$.

[1] Other link bandwidth allocations were tried and yield similar results.

IV. Simulation Results

Earlier works have shown that AF flows perceive a reduced congestion level [11] with respect to the actual network situation. This holds also for HTTP traffic, thanks to the differentiation of packet drop priorities, but causes a strong performance degradation of BE traffic: indeed the use of RIO-C entails that packets with low drop precedence can be dropped only when all the packets with higher drop precedence have been discarded; thus the BE packets suffer from a much higher drop probability, which causes much longer completion times.

A. Overprovisioning of HTTP Traffic

A.1 HTTP Traffic Only (Underloading)

In this case, both the DS and BE sources offer an average traffic load equal 50% of the overall offered load and half of the bottleneck is reserved to AF traffic. Therefore, the AF traffic volumes do not nominally exceed the established profile, except for short periods of time where congestion occurs due to packet bursts. This allows the investigation of network performance degradation as ρ grows.

Fig. 2 plots flow completion times: comparing the bottom plot (AF traffic) and the upper one (BE traffic) it can be immediately noted that AF outperforms BE traffic by two orders of magnitude. Moreover, for loads close to the line rate, BE flows are pushed toward starvation, whereas AF flow completion times remain quite low: on average, 90-packets-long flows achieve a throughput of 3,6 kbps in BE case versus 456 kbps in the AF one. Due to green-marked packet protection, AF traffic suffers a meager amount of packet drops with respect to BE. This behavior follows from the small ($< 0.1\%$) amount of AF packets marked as red for any ρ.

Further insight can be gathered from the comparison of AF versus BE performance depicted in Fig. 3, plotting packet drop percentages. Again, BE flows suffer from

drop probabilities that are one order of magnitude higher than the AF one. Whereas red marking is evenly distributed over a range of flow lengths, it can be seen that packet drops are not: BE shortest flows ($\pi = 1$) achieve the highest drop percentage for any ρ, while one-packet-long AF flows are completely protected against drops for $\rho < 0.8$, and only for $\rho > 0.8$ a small percentage of AF packets are dropped.

This looks like a counter-intuitive phenomenon, since one could expect that flows that are just one-packet-long should experience uncorrelated drops. As is well known, for flows sending less than 10 packets, the TCP congestion window is controlled throughout by the Slow Start algorithm, while longer flows are likely to reach a Congestion Avoidance phase. From a RED-AQM standpoint, longer flows are more likely to incur in early drop actions than shorter-lived flows; thus, long-lived flows usually experience a large early drop probability and this suggests that buffer occupancy is mainly due to longer flows (that are also more likely to benefit from Fast Recovery rather than incurring in retransmission timeout expiration). Fig. 4 confirms that this occurrence holds for any network load, as it can be seen that the early drop probability of longer flows is about twice as much as the one experienced by one-packet-flows. Notice also that the effectiveness of the RED-AQM decreases for larger values of ρ.

Conversely, for the AF traffic, long and short flows are substantially affected by the same drop probability; in this case, the benefits brought by RED-AQM are hardly distinguishable from the common DropTail performance.

A.2 HTTP Traffic with Background FTP Sources (Overloading)

In this experiment, we add 10 endless FTP sources in each cloud, which produce two main effects: bringing the overall network load close to the line rate and allowing the AF traffic to exceed its committed rate. This keeps the network congestion level almost constant, and the HTTP offered load modifies the percentage of more bursty traffic on the bottleneck link.

The increased amount of AF out-of-profile traffic (20% or more for any ρ) considerably influences the flow completion time, as shown in Fig. 5. Degradation is even larger looking at the BE performance figure. Significant AF packet drops occur even at low HTTP loads, and show linear dependence on ρ (Fig. 6); BE shorter flows are again the most penalized. In this scenario, a small percentage (0.15%) of HTTP-AF green packets are dropped, raising to 0.5% when taking into account the FTP traffic as well; furthermore, green-packet drops are almost entirely due to buffer overflow, indicating that RED-AQM is unable to manage the queue: seldom, it may happen that sudden packet bursts cannot be handled by router queues and green packets are dropped due to full buffer occupation. This might have been avoided if bursty traffic fluctuation had not been filtered out by RED-AQM mechanism. Indeed, the early drop mechanism is clearly more effective for smooth flows than for bursty traffic: for larger values of ρ, which correspond to larger burstiness, the percentage of early drop decreases, as shown in Fig. 7.

B. Underprovisioning

In this scenario we fixed the overall load to $\rho = 0.85$, while its per-cloud fractioning varies, in order to investigate the DiffServ reactions to a traffic volume exceeding the contracted SLA_{50} profile. The α_{AF} threshold discriminating between in- and out-of-profile packets is $\alpha_{AF,th} = 0.589$. No background FTP sources are present.

Fig. 8 and Fig. 9 respectively depict the completion times and the packet drop percentages achieved on average by AF and BE flows as a function of the AF HTTP fraction α_{AF} of the offered load (as usual $\alpha_{AF} + \alpha_{BE} = 1$). As expected, when the AF traffic load exceeds $\alpha_{AF,th}$, completion times gradually increase, and so does the drop percentage: as a growing portion of AF traffic is red marked, RED-AQM tries to limit the opportunistic AF transmissions via early drop – which is not anyway the main cause of dropped packets.

BE performance can be interpreted as follows. At low α_{AF}, when the HTTP traffic mainly comes from the BE cloud, the protection granted to AF packets forces a sizable number of BE packets to be discarded. Conversely, as α_{AF} grows, a reduced BE traffic volume better exploits the excess bandwidth, gaining from the competition with too aggressive AF flows: this eventually entails a reduction of both the BE drop percentage and completion time. Early drops for the latter scenario are shown in Fig. 10 and they illustrate an interesting phenomenon: packets from shorter AF flows (i.e., $\pi = 1$) are less likely to incur in early drops because of their lack of correlation, while longer flows can potentially reach long windows and drop a windowful of data. As for early drops for BE traffic, the relatively larger loss rate at opposite ends of the AF load spectrum can be justified noting that, at low AF loads, the BE traffic is high and thus the high-drop-preference queue is often full; at high AF loads, although the BE traffic is small, it shares the high-drop-preference queue with out-of-profile, red-marked AF traffic, which brings the drop probability higher than it would have been if only the BE traffic were present.

Finally, we observed that flows sending up to 6 packets exhibit substantially identical performance: in order to explain this phenomenon, we should consider specific TCP dynamics. Indeed, TCP infers that a segment is lost either by triggering the Fast Recovery algorithm or, least

desirably, when the retransmission timer expires. However, Fast Recovery cannot be triggered unless the congestion window is at least four segments large and the flow is long enough to allow the congestion window to grow to such limit. TCP congestion window constraint is the main cause of the unfairness toward such short-lived flows: the unfairness in resource utilization depending on flow length is a consequence of the increased probability for red-marked packets of smaller flows to incur in RTO.

V. Conclusions

The results obtained in this simulation study partially confirm the ability of the DiffServ model to offer QoS to IP flows. Its unquestionable pros –aggregated flow handling, scalability, incremental deployment, complexity limited to the edges of the network and the decoupling of resources provisioning from the forwarding path– justify the growing interest; furthermore, the chance of various practical implementations paves the road to research and extension of the present service classes – or to the specialization of services within the same class.

Conversely, we encountered objective difficulties to quantify the effective offered QoS level. This seems to suggest that, first of all, SLA contracts should avoid to state explicitly how the excess bandwidth should be shared; indeed, it seems as though the ISPs are required to perform intensive monitoring to align the pricing policy with the actually delivered service.

Moreover, a crucial research topic is represented by an efficient active queue management technique, possibly leading to an alternative to RED-AQM to address the unfairness problem. The feeling is that as parameter tuning in RED-AQM remains questionable, a single misconfigured router can determine the performance of the whole network; specifically, there is a trade-off between improving completion time performance of short and long flows at once: the former benefit from shorter queues (decreasing thus the queuing delay) while the latter from longer queues (since a reduced drop rate entails a reduction of RTOs).

Finally it should be noted that, rather than resulting in network decongestion, repeatedly forcing a short-lived flow in RTO results mainly in excessive flow penalty, whereas transmission would otherwise end after sending a few more segments. Since today's Internet traffic is heavily represented by short-lived connections, the excessive TCP unfairness among flows with different lengths cannot be disregarded; however, solutions to this problem are not straightforward, since they may lead to consider specific TCP congestion control changes.

References

[1] R. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, RFC 1633, Jun. 1994.

[2] D. Black, S. Blake et al., *An Architecture for Differentiated Services*, RFC 2475, Dec. 1998.

[3] V. Jacobson, K. Nichols, K. Poduri, *An Expedited Forwarding PHB*, RFC 2598, Jun. 1999.

[4] J. Heinanen, F. Baker et al., *Assured Forwarding PHB Group*, RFC 2597, Jun. 1999.

[5] M. Mellia, I. Stoica, H. Zhang, *Packet Marking for Web traffic in Networks with RIO Routers*, Globecom 2001, San Antonio, Texas, Nov. 2001

[6] S. Sahu, D. Towsley, J. Kurose, *Quantitative Study of Differentiated Services for the Internet.* IEEE Global Internet 99 Conference, Rio de Janiero, pp. 1808-1817, Dec. 1999

[7] W. Feng, D. Kandlur, D. Saha, K. Shin, *Understanding TCP Dynamics in a Differentiated Services Internet*, IEEE/ACM Transactions on Networking, Vol. 7, pp. 173-187, Apr. 1999

[8] W. Feng, D. Kandlur, D. Saha, K. Shin, *Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet*, IEEE/ACM Transactions on Networking, Vol. 7, pp. 685-697, Apr. 1999

[9] *ns* Network Simulator, Available at *http://www.isi.edu/nsnam/ns/.*

[10] W. Fang, N. Seddigh, B. Nandy, *A Time Sliding Window Three Color Marker*, RFC 2859, Jun. 2000.

[11] D. Clark, W. Fang, *Explicit Allocation of Best Effort Packet Delivery Service*, IEEE/ACM Transaction On Networking, Vol. 6, pp. 362-373, Aug. 1998

[12] A. Feldmann, J. Rexford, R. Caceres, *Efficient Policies for Carrying Web Traffic over Flow-Switched Networks*, IEEE/ACM Transactions on Networking, Vol. 6, pp. 673-685, Dec. 1998

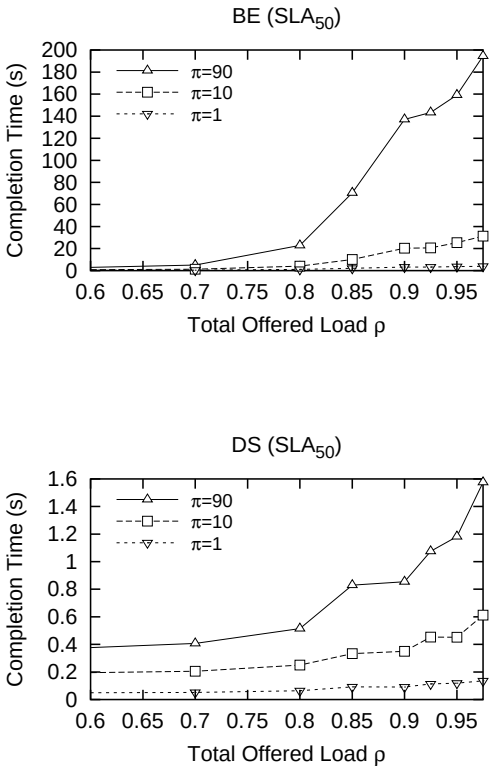
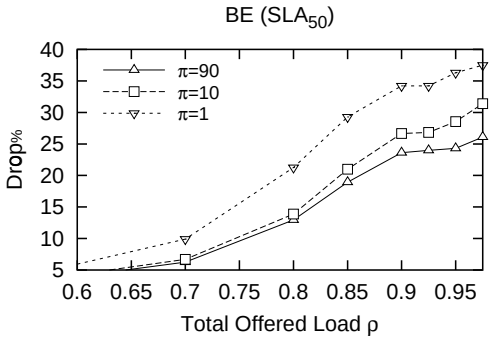


Fig. 2. Overprovisioning (Underloading): Completion Time



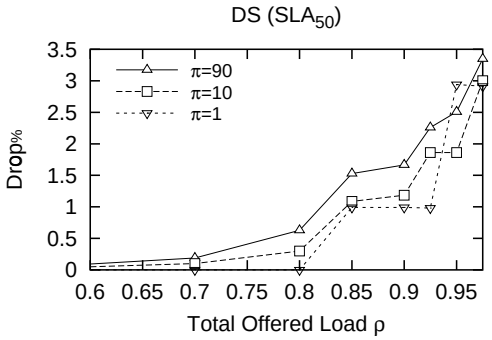
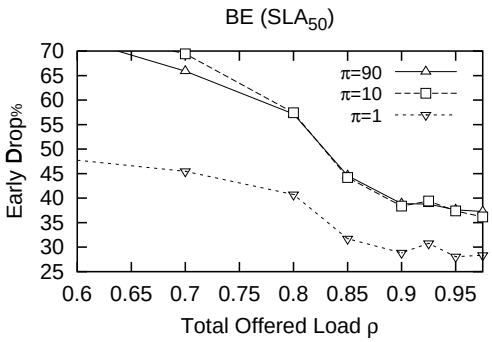


Fig. 3. Overprovisioning (Underloading): Packet Drop Percentage



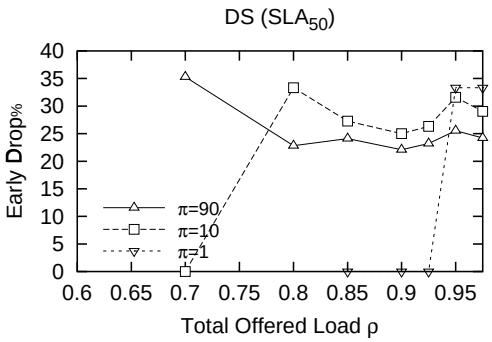
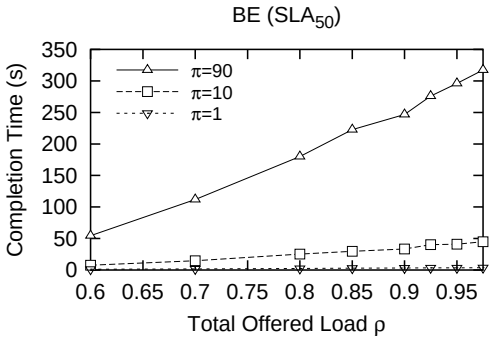


Fig. 4. Overprovisioning (Underloading): Early Drop Fraction



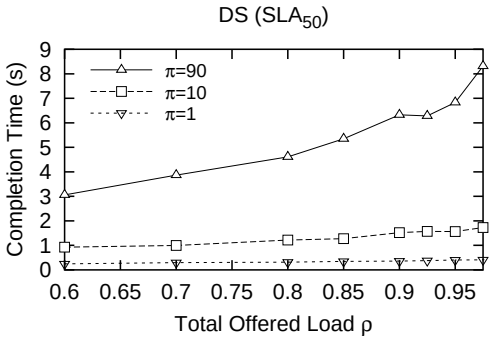
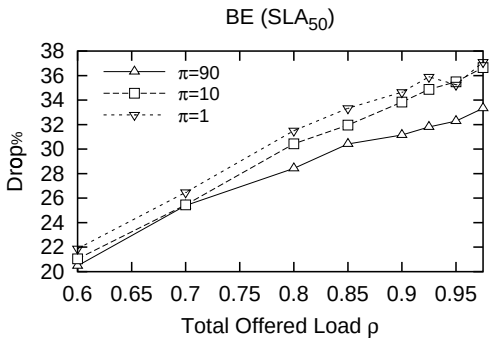


Fig. 5. Overprovisioning (Overloading): Completion Time



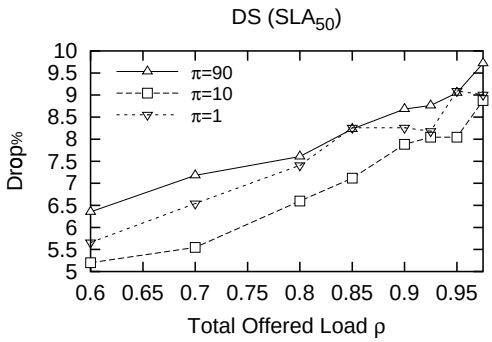
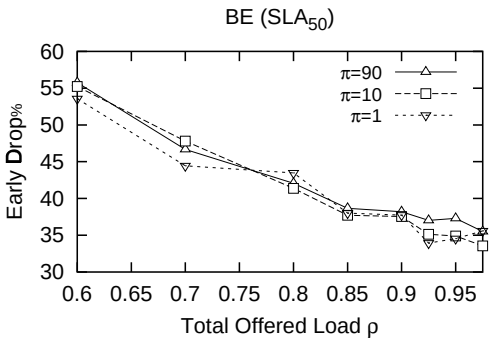


Fig. 6. Overprovisioning (Overloading): Packet Drop Percentage



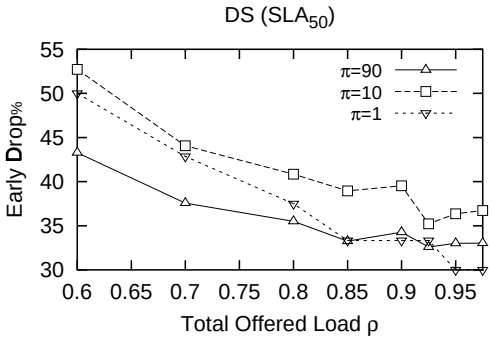
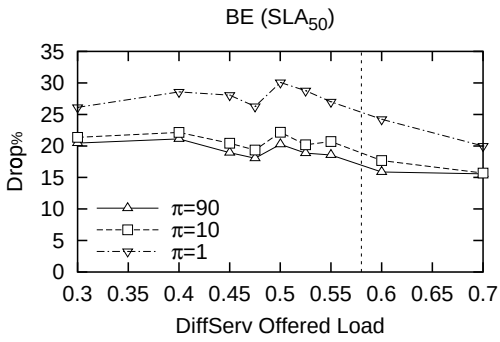


Fig. 7. Overprovisioning (Overloading): Early Drop Fraction



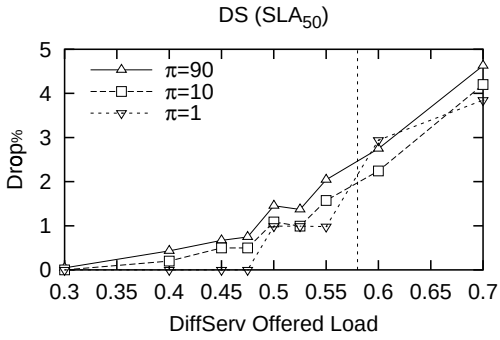
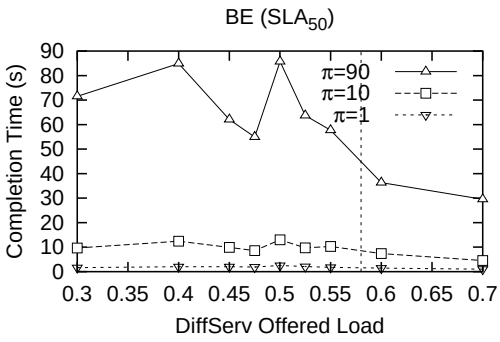


Fig. 9. Underprovisioning: Drop Percentage



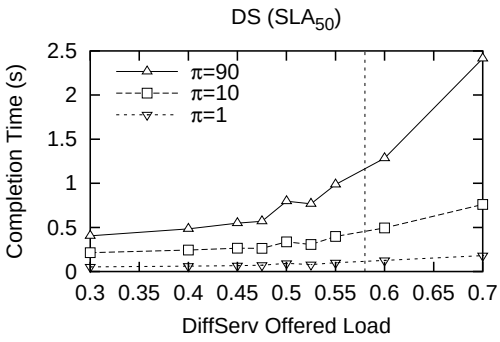
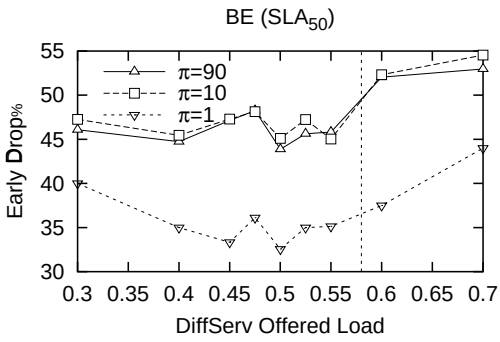


Fig. 8. Underprovisioning: Completion Time



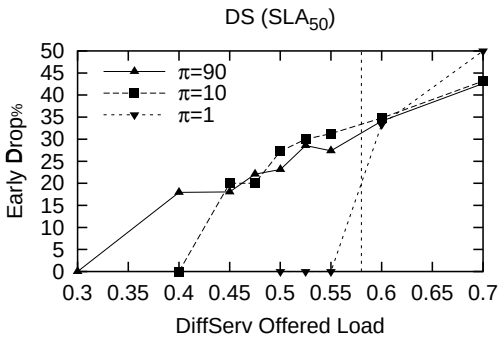


Fig. 10. Underprovisioning: Early Drop Fractioning