Traffic Priorization and Differentiation with Active Queue Management

Uwe Walter, Klaus Wehrle
Institute of Telematics, University of Karlsruhe
76128 Karlsruhe, Germany
Phone: +49 721 608 6402, Fax: +49 721 388097
{walter,wehrle}@tm.uka.de

Abstract

Active queue management is a powerful tool to improve network performance, especially with respect to TCP's flow and congestion control mechanisms. In addition, it is also suitable to realize priorization between distinct traffic aggregates, e.g. within the Assured Forwarding PHB group.

In this paper the authors first summarize and underline the benefits using active queue management mechanisms for realizing several service classes using a common queue. Furthermore different realization variations are presented and evaluated on the basis of extensive simulative analyses. For example it is possible to reach distinct levels of differentiation by using varying approaches for the development of virtual queues. The influences of different variations of the RED-algorithm (among others Gentle-RED and Adaptive-RED) are evaluated as well.

Finally the results have been used by the authors as a basis for the realization of two actual Differentiated Services per-hop behaviors, namely the Assured Forwarding PHB group and the Limited Effort PHB.

1 Motivation for statistical QoS guarantees for adaptive applications

Network services that are able to ensure deterministic guarantees for quality of service (QoS) parameters like throughput, delay and packet loss are on the one hand very useful for a lot of applications, but, on the other hand, pose a lot of restrictions on the characteristics of the corresponding data flows. For example, the draft of the Virtual Wire Per Domain Behavior (PDB) [7] specifies that all packets, which exceed the committed data rate have to be unconditionally discarded, even when unused bandwidth would still be available.

Therefore, such network services are not suited for applications that show fluctuations in their bandwidth demands and especially adaptive applications, which can adjust to the available bandwidth by according mechanisms [6, 10]. The main reason for the development of these adaptive applications has been the lack QoS-capable network services in the present Internet, which generated the demand for efficient mechanisms that could help to adapt to changing network conditions (e.g. by choosing appropriate encoding algorithms [13]). However, the margins within these mechanisms operate successfully are not very large, especially regarding the achieved user satisfaction.

This can be illustrated by having a look at the example of Internet radio transmissions. To compensate fluctuations of the available bandwidth and occurring packet losses, streaming applications (like the realplayer for example) buffer the received data for a certain amount of time, so they are able to play it back delayed, but as undisturbed as possible. The higher the data rate of the concerned application and the more packet losses occur, the bigger the needed playback buffer must be made in order to compensate all disturbances. As a result of this current absence of quality of service support in the Internet, the deployment of such applications is mostly limited up to a data rate of 512 kbps (as a rule of thumb). Nevertheless, depending on the network load, it is sometimes still not possible to compensate or avoid all disturbances.

If, on the other hand, a certain amount of bandwidth would be guaranteed for these applications, it would be possible to achieve higher throughputs. Through this, it would be feasible to reserve the appropriate bandwidth for the minimum quality that would be acceptable and the application could further try to increase the achieved throughput with the help of its adaption mechanisms and thereby increase the playback quality.

Because of these reasons, the realization of a QoS-supporting network service that guarantees a certain minimum throughput but allows the utilization of more bandwidth if available, will be investigated in the following. As a basis for this quality of service support for adaptive applications the Assured Forwarding Per Hop Behavior, developed by the Internet Engineering Task Force (IETF) will be used.

2 The Assured Forwarding Per Hop Behavior

The Assured Forwarding (AF) Per Hop Behavior as specified in [1] defines N independent AF classes (AF groups). Within these groups, every packet is assigned to one of M different levels of drop precedences – shortened as AF_{ij} with $1 \leq i \leq N$ and $1 \leq j \leq M$. Twelve different DiffServ codepoints (DSCPs), divided to four AF classes ($N = 4$) each with three different drop precedences ($M = 3$) AF_{x1} up to AF_{x3} have been standardized in RFC 2597 for common usage. The packets with the lowest drop precedence AF_{x1} must not be discarded more frequently in the statistical mean than packets with the "priority" AF_{x2} belonging to the same AF class AF_{x}. If only two different drop precedences are implemented, the same drop probability must be assigned to the priorities AF_{x2} and AF_{x3}, otherwise the above relation also applies to the drop precedences AF_{x2} and AF_{x3}.

Between different AF classes, there is no such dependency. On the contrary, the definition of the Assured Forwarding PHB in [1] even demands that data flows of different AF classes may not be jointly forwarded and therefore must not be treated together as an aggregated stream. Each AF class obtains a configurable fraction of the available resources, e.g. bandwidth or queue buffer. RFC 2597 does not prescribe a particular implementation, but poses the condition, that it should be able to provide each class with the assured resources, independent of the fact, if this is measured over small or large time scales. In addition, the deployment of active queue management algorithms is demanded to help in achieving the best possible utilization of the available resources. The strategy for the division of remaining spare resources between the competing aggregates is left as an implementation decision. For these reasons, the Assured Forwarding Per Hop Behavior represents a generic, basic structure for the differentiation of traffic in separate aggregates with different behavior.

In section 4 the realization of the Assured Forwarding Per Hop Behavior with the means of a flexible, building-block oriented architecture for protocol extensions is investigated. At this, the implementation of a single AF class is observed at first, followed by the separation of the four AF classes. On the basis of the desired attributes, the necessary building blocks (called modules) are selected and different realization options are compared and evaluated in simulations. Since the Assured Rate Per Domain Behavior (PDB) has a crucial influence on the quality of service attributes of the Assured Forwarding PHB, it will be presented in the following to allow the evaluation of the combined implementation of both forwarding behaviors afterwards.

3 The Assured Rate Per Domain Behavior

The Assured Forwarding Per Hop Behavior is used in the construction of the Assured Rate Per Domain Behavior [11]. This was designed for aggregates that demand an assured bandwidth but do not need guarantees for upper bounds of delay and delay variations (jitter). Therefore, the Assured Rate PDB is suitable for the application class of adaptive applications that need statistical bandwidth guarantees as described in section 1, e.g. for the construction of Virtual Private LANs (VPNs), as long as no intolerant interactive real-time applications shall be used via these VPNs.

The Assured Rate PDB uses a single AF class for packet forwarding. As long as an aggregate does not exceed its assured data rate, also denoted as *Committed Information Rate* (*CIR*), its packets are marked with the drop precedence (priority class) AF_{x1} – also denoted as "green" in Assured Rate. The compliance of an aggregate to its committed information rate can be verified in a domain border node by metering and averaging the data traffic over a certain time interval T_1, for example with the usage of a token bucket meter configured with the CIR as token rate and the *Committed Burst Size* (*CBS*) as bucket depth.

Non conforming packets (that exceed the configured committed rate) are not discarded immediately, but marked as “yellow” or “red” instead and forwarded with a higher drop precedence (corresponding to the AF priorities AF_{x2} respectively AF_{x3}). Hereby, they can try to take advantage of remaining resources that are unused by other aggregates. It is left as a free decision to the network operator, how exactly the non-conforming packets are treated. For example, they could all be unconditionally marked as yellow or red or as another option fed into a second traffic meter and averaged over a second time interval T_2 to further differentiate them in this second stage into separate levels of non-conformance and mark them accordingly as yellow or red.

To be able to actually assure the minimum bandwidth CIR, the specification of the Assured Forwarding PHB is extended by the demand that in normal operating conditions *no* packet loss should occur in the priority class AF_{x1}. Nevertheless, the Assured Rate (AR) PDB definition only mentions a statistical bandwidth guarantee, since packet losses can not be excluded completely. As the Assured Forwarding PHB [1] explicitly recommends an active queue management algorithm, like *Random Early Detection* (*RED*) or similar mechanisms, the parameters of this algorithm must be chosen carefully to avoid green packets from being discarded. Consequently, the active queue management is hereby virtually deactivated for green marked packets. Yellow and red packets shall be continued to get discarded early, if an oncoming congestion condition is detected.

The AR PDB leaves the decision for the actual procedure, how metering and marking is performed within the defined framework, to the service provider’s implementation. That’s why all deployed mechanisms and their parameters must be put down in a *Traffic Conditioning Specification* (*TCS*) that becomes contractual between the user and the network service provider. In this part of the *Service Level Specification* (*SLS*) it is agreed upon the traffic profile, together with the procedures for its metering, conformance checking and the treatment of excess traffic. On the other hand, the SLS comprises all technical aspects of a *Service Level Agreement* (*SLA*) and determines, along with other influencing factors (e.g. the topology of the DiffServ domain) the configuration of a per domain behavior. For example, in the Assured Rate PDB, the agreed data rate in the SLS determines the fraction of bandwidth, that must be assigned to the used AF class, which again determines the configuration of the scheduling policies deployed for serving the queues of the different AF classes and also the other DiffServ network services. The SLS, together with the economical and political aspects of a service agreement, finally forms the mentioned Service Level Agreement.

In the following, the realization of the forwarding behaviors Assured Forwarding PHB and Assured Rate PDB will be discussed. In this context, it will be evaluated which implementation options exist, how they can be deployed with a module-based framework and which possibility achieves the desired quality of service level.

4 Realization of the AF-PHB and the AR-PDB

As it showed during the evaluations for this work, it is not possible to completely separate the realization of the AF PHB from the AR PDB, as it might be feasible with other DiffServ classes (e.g. the Virtual Wire PDB and its underlying Expedited Forwarding PHB), since the requirements of the Assured Rate PDB represent a strong intensification of the Assured Forwarding PHB, as it will be further illustrated in section 4.5.

Therefore, the realization of the two forwarding behaviors AF and AR will be observed together with the main target of fulfilling the requirements of the Assured Rate PDB. For a better understanding of the special demands of the Assured Rate PDB, it is helpful to start by observing the deployed traffic conditioning mechanisms and, based on those, continue to investigate the differentiated forwarding of the categorized packets by the DiffServ mechanisms of the Assured Forwarding PHB.

4.1 Usage control of the Assured Rate PDB

The quality-supporting network service Assured Rate PDB offers a statistical guarantee for a minimum bandwidth R_{CIR}. However, it shall be explicitly allowed to exceed this rate, with the downside that excessive packets then get subject to a higher drop precedence. Up to a chosen maximum bandwidth R_{PIR} (*Peak Information Rate – PIR*) a lower drop probability should be used as for exceeding the *PIR*. Since this network service was developed for adaptive applications, the achieved throughput of the considered AR-aggregate should settle down between the CIR and PIR.

Referring to the coloring scheme that has been described above, it is now the task of the usage control in the first DiffServ-capable network node to meter the packets of an Assured Rate data flow and mark them accordingly as follows:

- All packets arriving with a lower rate than the minimum rate, are marked as green, which corresponds to the drop precedence AF_{x1}. Thus, these AF_{x1}-packets may not be discarded. Since it is not always possible to guarantee this for sure, the Assured Rate PDB only offers a statistical network service.

- If the minimum bandwidth is exceeded, but the packets still arrive slower than the maximum rate (PIR) specifies, they are categorized as yellow and are marked with the DSCP of the AF_{x2} drop precedence. No guarantee is given for the forwarding of these packets, but it is more probable for them to get transmitted successfully than for the red packets.

- All packets that exceed the maximum rate (PIR) are marked red and therefore belong to the priority class AF_{x3}. As soon as a congestion in a waiting queue occurs, red packets are dropped at first, before it is begun to discard packets of the other drop precedences.

These guidelines for the traffic conditioning mechanisms of the usage control can be realized in a module-based framework as depicted in figure 1. After the packets have been categorized by a Multifield-Classifier, they are metered in a first Token Bucket module, where their conformance to the Committed Information Rate is checked. If they actually are conforming, they get marked with the AF_{x1}-DSCP. If not, they are fed into a second Token Bucket, that verifies the Peak Information Rate. All packets that are categorized as conforming in this second test are marked as yellow (AF_{x2}) and the remaining packets that exceed both the CIR and PIR receive the red marking (AF_{x3}).

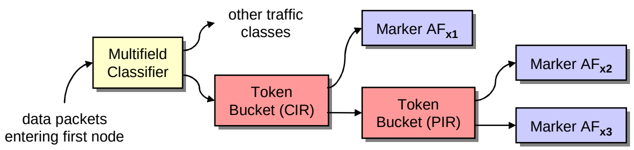


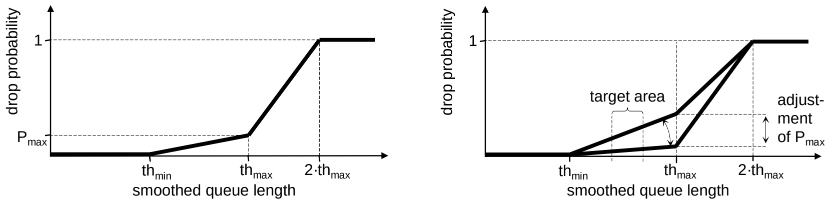
Figure 1: Schematic realization of the usage control of the Assured Rate PDB

4.2 Realization of an AF traffic class

In the following, at first the realization of a single AF class (AF_x) with three drop precedences AF_{x1} to AF_{x3} is investigated. In this context, the question arises, if separate waiting queues should be used for the different drop priorities. However, this question is virtually answered by the demand in [1] that within one AF class any packet reordering must be ruled out. As a result, all packets must be enqueued in first-in first-out order (*FIFO*) into *one* waiting queue. In a realization with multiple queues, packet-reordering cannot be avoided completely (or would at least require complex mechanisms to fight it). On the other hand, each separate AF class must use its own queue, otherwise packets of different AF classes would be enqueued in the same queue, which would be equal to treating them together as an aggregate, which is prohibited by [1]. Because of these reasons, exactly one waiting queue is used for each AF class.

The next problem arises from the necessary differentiation of the available drop precedences within an AF class. Since only one waiting queue can be used per AF class to avoid re-ordering, it is not possible to simply insert packets with different priorities into separate queues and realize the drop precedences by the use of corresponding scheduling policies. Since [1] recommends the deployment of an active queue management anyhow, this offers the opportunity to differentiate the packets by a weighted active queue management algorithm, like *Weighted-RED* (*WRED*).

Consequently, different options to realize the three drop precedences are presented and evaluated in the following. The partitioning of the



(a) Gentle RED

(b) Adaptive RED

Figure 2: Alternative versions of Random Early Detection

available bandwidth between the deployed AF classes is observed in section 4.6.

4.3 Alternative versions of Random Early Detection

Random Early detection is deemed as an effective mechanisms to perform active queue management [5]. By starting early to discard single, randomly chosen packets, strong increases of queue lengths can be reduced and imminent congestions can be avoided. However, some studies show that the chosen configuration parameters (th_{min}, th_{max}, P_{max}) have a crucial influence on the behavior of RED and its impact on the traffic that is traveling through the node. Inaccurate values for these parameters can lead to undesired performance of the mechanism [4, 9].

In order to make the behavior of RED more resistant to strong changes of the traffic condition and limit the impact of poor configurations, two variants of the RED algorithm have been developed:

- *Gentle RED:*

  If the configured thresholds for RED are chosen too low, this leads to very frequent and numerous packet drops, which is quite similar to the behavior and impact of the standard tail drop principle [9]. Also, in situations where the traffic condition changes too often, the behavior of RED can be too restrictive. Therefore, a modified RED version was proposed in [4], that in all shows a more "friendly" behavior and is called *Gentle-RED.* In this variation, *not all* packets are discarded, as soon as the smoothed queue length reaches the upper threshold th_{max}, but packets are only dropped with a linearly increasing probability, as shown in figure 2(a). Only when the smoothed queue length finally exceeds $2 \cdot th_{max}$, all packets are discarded unconditionally.

  Gentle-RED, as a whole, shows a more robust behavior when compared to the original RED, but implies longer average queue lengths, since the threshold for unconditional discarding of all packets is elevated. Nevertheless, Gentle-RED has been propagated very fast and is deployed in a lot of network nodes these days [12].

- *Adaptive RED:*

  In the Gentle-RED approach described above, there still exists the necessity to specify the parameters th_{min}, th_{max} and P_{max}. That's why a new technique has been developed in [3], where the drop probability P_{max} adjusts itself autonomously to the current traffic situation, by trying to retain the length of the waiting queue in a certain target interval[1]. The exact procedure of this automatic adaption is described in [3]. Figure 2(b) depicts the course of the drop probability depending on the smoothed queue length.

In the following analysis towards the differentiation of the drop precedences within an AF traffic class, both of these RED variants will be taken into consideration, as it will be evaluated which option leads to the best results.

4.4 Virtual queues by different counting procedures

After the introduction of variants of active queue management mechanisms, several approaches for differentiation of packets in the same queue will be presented in the following. As already mentioned in section 2, no packet re-ordering is allowed to occur within the aggregate flow of one AF class. Therefore, the needed differentiation is not accomplished *within* the queue, respectively by according scheduling policies,

---

[1] $\Big[th_{min} + \frac{th_{max}-th_{min}}{2} \pm 0.1 \cdot (th_{max} - th_{min})\Big]$

but instead by selectively dropping packets *before* they are even enqueued.

Differentiation can be achieved by a combination of the following two principles:

- *Virtual waiting queues:*

  Initially, another perception of the packets inside a queue can establish a differentiation. Through this, multiple *virtual* queues (namely one for each priority class) emerge out of a single real queue. For example, such a virtual queue can be generated by different packet counting procedures, as it will be presented in two approaches, following shortly.

- *Different drop criteria:*

  The usage of different drop criteria for the separate priority classes can also be used to achieve a differentiation of packets in a waiting queue. If the RED algorithm is deployed, this can be accomplished by different settings for the RED thresholds.

In the following, three procedures will be presented, that are based on the two principles described just above. The applicability of each of these approaches for the realization of an Assured Forwarding class will be evaluated in simulative analyses, described in section 4.5.

- *Single counter:*

  In this approach, three independent RED mechanisms are deployed for the realization of the three drop precedences within one AF class. However, all these RED instances share the same perception of the waiting queue, which is why the differentiation results only from the means of different thresholds, as it is depicted in figure 3. The outcome of this for the length of the considered queue $\mathcal{W}_i$ for priority class AF$_{ij}$ is the following (where L_p represents the length of packet p given in byte):

  $$L_{ij} = \sum_{p \,\epsilon\, \mathcal{W}_i} L_p \tag{1}$$

  The displaced alignment of the RED thresholds makes sure that at first only the lower priority AF$_{x3}$-packets get discarded, before the range for dropping the packets with medium priority is reached. If the queue length continues to rise, these packets are discarded, too. High priority AF$_{x1}$-packets are even enqueued, when all packets of the other drop precedences already get discarded.

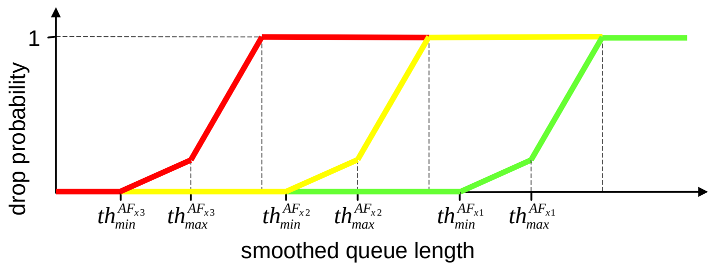


Figure 3: Differentiation of packets in one queue by separate RED mechanisms with different thresholds

- *Separate Counters:*

  Another possibility for the differentiation of packets with different drop precedences lies in the deployment of separate counters for the waiting queue. This equals a partitioning of the queue $\mathcal{W}_i$ in multiple virtual queues $\widetilde{\mathcal{W}_{ij}}$, where for each priority class AF$_{ij}$ only the packets p are counted that actually belong to this class AF$_{ij}$. The length of the virtual queue $\widetilde{\mathcal{W}_{ij}}$ can then be calculated as follows:

  $$L_{ij} = \sum_{\substack{p \,\epsilon\, \mathcal{W}_i \;\wedge \\ p \,\epsilon\, \mathrm{AF}_{ij}}} L_p \tag{2}$$

  One essential characteristic of this approach is, that the priority classes are differentiated relatively independent from each other, which means, that during the decision if a packet gets enqueued or must be discarded, only the packet counter of the according drop precedence AF$_{ij}$ is considered, while the counters of the other priority classes AF$_{il}$ $(l \neq j)$ have no influence.

- *Additive counters:*

  The independence of the different virtual waiting queues at the deployment of separate counters may be desirable in certain cases. However, in the course of realizing an AF group, there exists a tight relationship between the different drop precedences. If possible, the high priority packets AF$_{x1}$ shall always be forwarded and only if spare buffer space is still available, the lower priority packets AF$_{x2/3}$ should be enqueued, too. Therefore, in this case, the decision to enqueue or drop a packet should also take the situation of the other drop precedence classes into account. More precisely, only the packets of the affected drop precedence and higher priorities (that is lower drop precedences) should be considered:

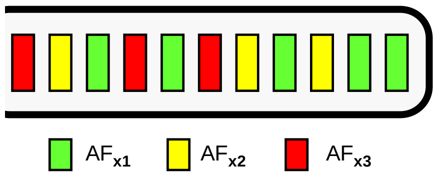
  $$L_{ij} = \sum_{k=1}^{j} \left(\sum_{\substack{p \,\epsilon\, \mathcal{W}_i \;\wedge \\ p \,\epsilon\, \mathrm{AF}_{ik}}} L_p \right) \tag{3}$$

  This principle of an additive counting procedure has already been discussed in the first approaches towards the Differentiated Services architecture, whereas at that time, the differentiation was limited to two priority classes (*Red with In and Out – RIO*) [8]. Within the scope of this work, a heuristics for choosing the RED thresholds of the RIO algorithm could get determined that can be expanded for the usage of additive counters [2]. More details of this can be found in [14].

Figure 4 compares the three presented counting procedures in an example of a waiting queue of the Assured Forwarding PHB. In the next section, the combinations of the presented possibilities for the realization of packet differentiation in one waiting queue and the three considered RED versions will be evaluated in simulative analyses.

4.5 Evaluation of the realization options

After the introduction of different techniques for the differentiation of packets in one waiting queue and several algorithms for active queue management, the results of evaluations of these mechanisms concerning their applicability for the realization of the Assured Forwarding PHB and the Assured Rate PDB will now be presented. In the scope of this work, numerous simulations have been conducted in [12] of which the most important results will be listed in the following.



(a) Real queue

Virtual queue	Single counter	Multiple counters	Additive counters
$\widetilde{\mathcal{W}}_{i1}$	11	5	5
$\widetilde{\mathcal{W}}_{i2}$	11	3	8
$\widetilde{\mathcal{W}}_{i3}$	11	3	11

(b) Length[1] L_{ij} of the virtual queues $\widetilde{\mathcal{W}}_{ij}$

Figure 4: Differentiation of packets in a virtual queue by different counting procedures

Concerning the provision of different drop precedences within an AF class, RFC 2597 defines that in the statistical mean, packets of the highest priority (and lowest drop precedence) AF$_{x1}$ shall not be discarded more frequent than packets of AF$_{x2}$ (compare section 2). Yet, in the specification of the Assured Rate PDB [11] this requirement is strengthened to the extend that AF$_{x1}$-packets ("green" packets) shall only be discarded in exceptional cases. These different definitions have essen-

---

[1] For simplification, it shall be assumed, that all packets are of the same length and L_{ij} represents the number of packets.

tial influence on the final realization of the AF PHB. Since in the scope of this work, the Assured Rate PDB should be realized and evaluated and since the AR specification is a tightening of the AF PHB, in the following, emphasis will be put on fulfilling the requirements of the AR PDB.

Thus, it is the aim of the following evaluations to determine, which combination of counting procedures and RED algorithms allows the best differentiation of the three drop precedences within an AF class in terms of the Assured Rate PDB. In addition, the question arises, if the same RED algorithm should be deployed for each of the three priority classes. All together, this results in $3^3 = 27$ different possible combinations, which could not all be extensively evaluated by simulations. Hence, some preselection has been done in the scope of [12], where preliminary investigations of the three RED versions and the counting procedures allowed the number of combinations to be reduced.

For example, it showed, as already assumed in the previous section that the separate counter method was not capable to implement the required priority for green packets as demanded by the Assured Rate PDB. Since each waiting queue was only considered on its own, it was indeed possible to partition the bandwidth in a fair manner, but not prioritized as desired in this case. The detailed results of these evaluations can be found in [12].

Furthermore, it was determined in the preliminary evaluations that the Gentle-RED algorithm features no significant advantages, which could not also be realized by RED or Adaptive-RED. Therefore Gentle-RED will also not be considered in the further analyses. As well, it will be abandoned to deploy the Adaptive-RED algorithm on green packets, since, if anyhow possible, they should not be discarded at all, as it has already been discussed. Following this requirement of the AR PDB, there is no sense in applying an adaptive RED mechanism for active queue management for this priority class, which would only try to keep the queue length in a predefined target range. Instead, the original RED algorithm will be deployed, configured with very high thresholds to minimize the dropping of green packets as far as possible.

Thus, the permutations (*RED*,*RED*,*RED*), (*RED*,*Adaptive-RED*,*RED*) and (*RED*,*Adaptive-RED*,*Adaptive-RED*) have been selected as algorithms for active queue management of the drop precedences AF_{x1} to AF_{x3} and have been combined and evaluated with the two counting procedures "Single counter" and "Additive counters". Furthermore, for each of the resulting six variants, three different threshold configurations have been investigated (Case I – III, see table 1).

Priority class	Additive counters			Single counter		
	Case I	Case II	Case III	Case I	Case II	Case III
AF_{x1}	$50 - 100$	$50 - 100$	$50 - 100$	$50 - 100$	$50 - 100$	$50 - 100$
AF_{x2}	$10 - 20$	$7.5 - 15$	$7.5 - 15$	$5 - 10$	$7.5 - 15$	$10 - 20$
AF_{x3}	$5 - 10$	$5 - 10$	$7.5 - 15$	$2.5 - 5$	$2.5 - 5$	$2.5 - 5$

Table 1: Threshold configuration of the active queue management (given in kilobyte)

The considered DiffServ-capable network node and the traffic generators have been configured in a way that green packets arrived exactly with the agreed minimum rate and consequently none of these packets should get discarded. In addition, there was enough bandwidth supplied that the yellow packets should also just about be able to get forwarded. Since the differentiation of a weighted active queue management mechanism is not always completely accurate, packet losses of green and yellow packets are nevertheless possible and must be taken into consideration. The aim of the evaluations was, to identify the realization variant, which would show the least loss of green packets and further on discard as few of the yellow packets as possible.

The results of the evaluations are listed in table 2. At first, it must be noticed that yellow and even green packets are discarded in every variant. However, the version using a single counter (Configuration III) in conjunction with the combination (*RED*,*Adaptive-RED*,*RED*) featured the best results.

Overall considered, the variant with single counters showed in most cases a superior performance over the version with additive counters. Also, the average queue length of the AF_x-waiting queue was smaller than with the other counting procedures [12]. The reason for this, apparently lies in the fact that the complete waiting queue is considered

Test series	Measured packet loss rate [%] with the RED versions:					
	RED/Adap./RED		RED/Adap./Adap.		RED/RED/RED	
	AF_{x1}	AF_{x2}	AF_{x1}	AF_{x2}	AF_{x1}	AF_{x2}
Additive Count. (I)	0,475	4,8	0,375	5,9	3,75	4,07
Additive Count. (II)	0,55	4,5	0,4	5,2	0,4	5,3
Additive Count. (III)	0,85	6,9	0,7	9,0	0,7	9,4
Single Count. (I)	0,075	7,9	0,1	6,9	0,01	6,0
Single Count. (II)	0,075	4,3	0,05	4,6	0,05	3,8
Single Count. (III)	0,05	3,1	0,075	3,5	0,075	2,8

Table 2: Measured packet loss rate of the priority classes $AF_{1/2}$ with the different RED versions (given in percent)

as basis for each decision to enqueue or discard a packet and not only the virtual queue as it is the case with the other version of counting procedures.

The advantage of the combination (*RED*,*Adaptive-RED*,*RED*) can be explained as follows. During the separation of the three drop precedences, according to AR/AR, green packets should always get forwarded. If there is more spare buffer available, this remaining space (or similar: the remaining bandwidth) should be used for yellow packets. When the length of the queue reaches a maximum level, packets should not be discarded by a simple tail-drop mechanism, but an active queue management algorithm should be deployed instead, with its inherent advantages. Since the resources that are available for yellow packets are not protected by the admission and usage control, their actual amount can vary. This means, that changing conditions are especially common in the AF_{x2} priority class and should be adequately compensated within the forwarding behavior by the usage of adaptive mechanisms, which is exactly the reason, why the deployment of an *adaptive* queue management algorithm can be recommended. Extending the usage of these adaptive mechanisms on the treatment of red packets is not useful, since this adaption stage could potentially interfere with the active queue management of the AF_{x2} drop precedence and decrease the overall performance.

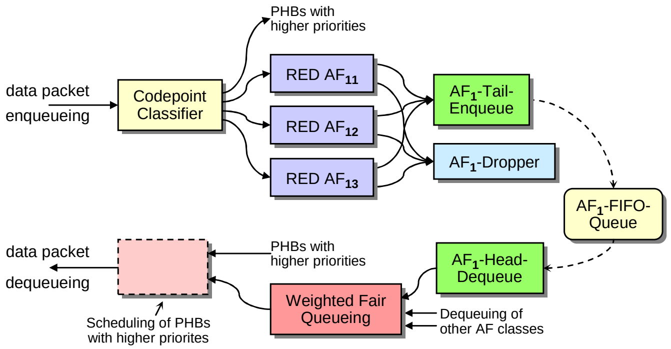


Figure 5: Schematic realization of the Assured Forwarding PHB scheduling

Thus, the deployment of a standard FIFO waiting queue (single counter) with the RED mechanisms (*RED*,*Adaptive-RED*,*RED*) for drop precedences AF_{x1} to AF_{x3} is recommended for the realization of an Assured Forwarding class. Figure 5 depicts a schematic representation of such an implementation (using a module-based framework for protocol extensions).

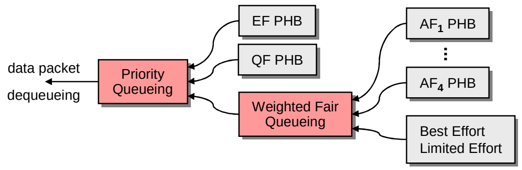
4.6 Differentiation of several AF classes

After the realization of an Assured Forwarding class, regarding the best possible conformance to the AR PDB specification, it will now be evaluated, how multiple parallel AF classes should be implemented. This will be done with a particular focus on the bandwidth partitioning between the separate AF classes. The target is a configurable amount of allocated bandwidth to the different classes, where remaining unused transmission capacity should be at first divided between the other AF classes. Only, if there is still bandwidth available, after it has been offered to all deployed AF classes, other network services, e.g. Best Effort (BE) or Limited Effort (LE) [15] should be able to make use of it.

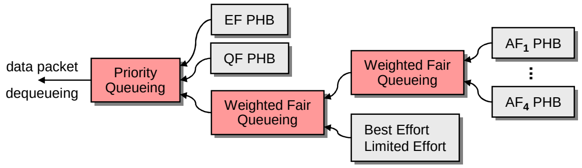
In this context, the question arises, how the Assured Forwarding classes should be integrated with the other network services (Expedited Forwarding (EF), BE/LE) into a combined framework. Since the forwarding behavior EF is of high priority, it is mostly served by a strict priority scheduling. The division of the remaining bandwidth must not necessarily follow a priority order, because the forwarding behaviors AF/AR and BE/LE do not demand strict bounds of the least possible delay. The only requirement is a weighted division of the available bandwidth, which is, why the scheduling strategy Weighted Fair Queuing (WFQ) is deployed. Hereby, it is possible to assign certain bandwidth shares to the AF/AR and Best Effort (including Limited Effort) forwarding behaviors.

In such a way, there exist two possibilities to partition bandwidth among the single AF classes that are depicted in figure 6. On the one hand, only one WFQ module could be used to divide the bandwidth between AR/AF and BE/LE, which would therefore have to serve five network services as it is shown in figure 6(a). On the other hand, it would be possible to deploy a separate WFQ module, that would partition its bandwidth share exclusively between the AF classes (see figure 6(b)). In both cases, it would be expected, that the desired bandwidth assignment would be reached, if all service classes would consume their share completely. This could be confirmed in the scope of this work by evaluations in [12] and also in [15].

If, however, one network service class (especially an AF class) would not use up all of its assigned bandwidth, it gets interesting, how the unused capacity gets redistributed. This has been evaluated in simulations with the result that the deployment of a separate WFQ module achieves the desired partitioning [12]. For this purpose, two AF classes (according to section 4.5) have been modeled and served by the two realization options depicted in figure 6. A minimum bandwidth of 10 Mbps and a maximum bandwidth of 15 Mbps were assigned to each AF class, which was actually completely utilized by AF class AF_2. The offered load to AF_1 was increased step-by-step from 5 Mbps up to 20 Mbps, so that at first the assigned bandwidth was not fully consumed. The offered load to the remaining network services EF and BE was explicitly higher than the bandwidth available to them. Consequently, they were able to make use of the unused bandwidth of AF_1.



(a) One WFQ module



(b) Separate WFQ modules

Figure 6: Different approaches for bandwidth division between several AF classes

Figure 7 shows the measured results that confirm the assumptions. At the deployment of only a single WFQ module, the remaining bandwidth gets equatable divided between *all* served network services. Since Best Effort was a part of them, it could also directly take advantage of a share of the unused capacity. Because of this reason, the AF_2 class could not use *all* of the spare bandwidth of AF_1 and only achieves a throughput of 12.35 Mbps to 13.39 Mbps. On the other hand, at the deployment of a separate WFQ module the unused capacity of AF_1 is in the first place re-distributed *within* the directly served network services, that is, between the other AF classes. Thus, in this case, AF_2 reaches the maximum possible throughput of 15 Mbps. The exact topology and details of the test setup can be found in [12].

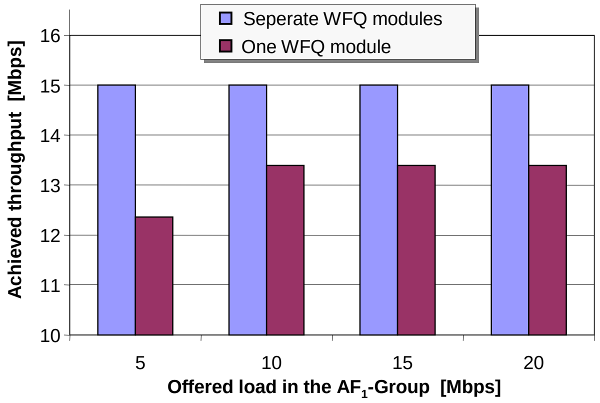


Figure 7: Comparison of the achieved throughput of AF$_2$ with one, respectively separate WFQ module(s) between the AF classes

Resulting from this evaluations, in the following, a separate WFQ module will be used to partition the bandwidth between the different AF classes (as depicted in figure 6(b)).

5 Evaluation of the statistical bandwidth guarantee

After the realization of multiple AF classes with their different drop precedences, some evaluations will be presented in this section concerning the statistical bandwidth guarantee provided by the Assured Rate PDB. For this purpose, numerous simulation runs with different DiffServ domains and varying traffic conditions have been performed within the scope of this work [12]. The main target of this research was, to verify, if the realization of the AR/AF forwarding behaviors that has been developed in section 4, would be able to achieve the desired statistically guaranteed throughput of the AR PDB and up to which extend, remaining unused capacity could be exploited.

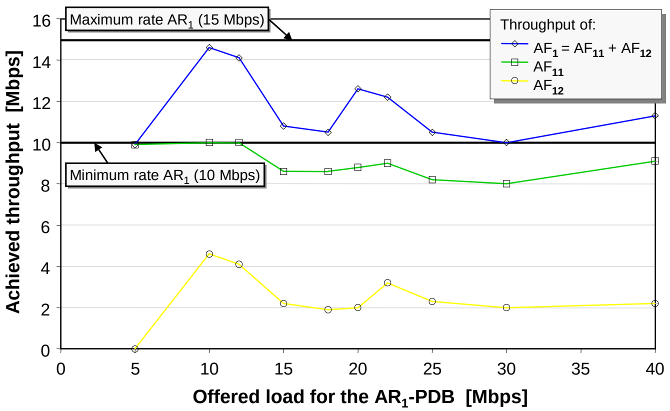


Figure 8: Achieved throughput of one AF class depending on the offered load

Since the performed evaluations have been quite extensive in respect of their configuration and the different traffic generators deployed, again, it must be referred to [12] for detailed information about them. As in the preceding experiments, a minimum bandwidth of 10 Mbps and maximum bandwidth of 15 Mbps have been assigned to the different AF classes. The generated traffic, representing AF traffic, has been adaptive (TCP data flows), which means, it could adapt to the condition of the network's load.

The achieved throughput of one AF class depending on the offered load is shown in figure 8. The depicted course corresponds as far as possible to the behavior that could be observed in most simulation runs. Again, it shall be repeated at this point, that the Assured Rate PDB offers only a statistical quality of service class, which should be utilized by adaptive applications. Thus, a strict deterministic behavior (as it might be monitored for example within the Expedited Forwarding forwarding behavior) can not be anticipated.

The course of the achieved throughput shows that, essentially the assigned minimum bandwidth of 10 Mbps could be guaranteed. Equally, the maximum bandwidth of 15 Mbps was almost reached. However, if

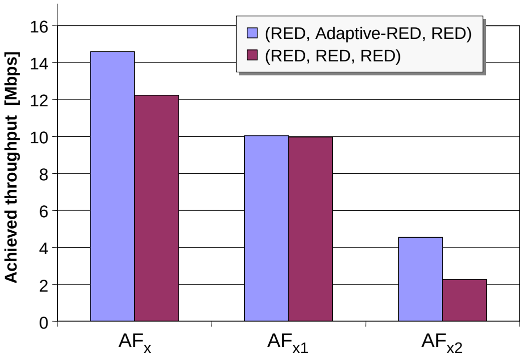


Figure 9: Achieved throughput of one AF class in the AR PDB

the offered load, that is the traffic amount sent by the traffic source, exceeds the maximum bandwidth, occurring packet losses are inevitable and because of the adaptive mechanisms of the Transmission Control Protocol, the achieved throughput decreases. This behavior has been observed in all larger scenarios in [12].

Thus, the developed realization of the Assured Rate PDB fulfills its requirements. First of all, that is the statistically guaranteed minimum bandwidth (CIR). Further more, it can be noticed, that it is indeed possible to use additional capacity, but this approach is actually limited by the configured maximum bandwidth (PIR). All attempts to utilize more capacity, inevitably lead to higher loss rates and, therefore, to a descent of the effective throughput (*goodput*) because of the adaptive measures of TCP.

In some of the larger scenarios, just mentioned, further evaluations for the validation of the implementation decisions in terms of different RED combinations of section 4 have been performed. The results confirmed the choice of the (*RED*,*Adaptive-RED*,*RED*)-combination, as it was the only one that nearly achieved the maximum throughput of 15 Mbps (see figure 9). The alternative (*RED*,*RED*,*RED*)-version did only achieve a maximum throughput of 12.2 Mbps, whereas it was clearly visible that this shortfall was caused by the active queue management of the yellow AF$_{x2}$-packets.

6 Conclusion

An adequate quality of service support for adaptive applications is an essential demand for the next generation Internet. The Assured Rate Per Domain Behavior on the basis of the Assured Forwarding Per Hop Behavior seems to satisfy these demands. But due to the nature of PHB and PDB definitions, especially with the absence of a definition of concrete mechanisms, respectively algorithms, the resulting behavior is heavily influenced by implementation details.

In this paper, we have discussed and compared various aspects of implementing the Assured Rate PDB and their influence on the resulting service quality. We enumerated three different algorithms for active queue management, three different techniques for the differentiation of flows in virtual queues and investigated the implications of each combination on the resulting QoS. Furthermore, we evaluated the flexibility of the Assured Rate PDB and its ability to use more bandwidth than the assured Committed Information Rate up to a configurable Peak Information Rate.

The developed realization option with single queue length counters and a (*RED*,*Adaptive-RED*,*RED*)-combination for active queue management and the results that have been observed during the performed simulations should be helpful for the planning, implementation and deployment of an Assured Rate network service as specified in the IETF's DiffServ architecture.

References

[1] Fred Baker, Juha Heinanen, Walter Weiss, and John Wroclawski. Assured Forwarding PHB Group. RFC 2597 (Proposed Standard), June 1999.

[2] Roland Bless and Klaus Wehrle. Evaluation of Differentiated Services using an Implementation under Linux. In *Proceedings of the 7th IFIP Workshop on Quality of Service, London*. IEEE, June 1999.

[3] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management, August 2001.

[4] Sally Floyd. RED: Optimum functions for computing the drop probability? http://www-nrg.ee.lbl.gov/floyd/REDfunc.txt, October 1997.

[5] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[6] H.-P. Gundelwein. Entwurf und Analyse adaptiver Verkehrsformungsmechanismen in DiffServ-basierten Endsystemen. Diploma Thesis, Institute of Telematics, University of Karlsruhe, 2000.

[7] Van Jacobson, Kathleen Nichols, and Kedar Poduri. The Virtual Wire Per-Domain Behavior. Internet-Draft ftp://www.packetdesign.com/ietf/vw_pdb_0.pdf, July 2000.

[8] Van Jacobson, Kathleen Nichols, and Lixia Zhang. A Two-bit Differentiated Services Architecture for the Internet. RFC 2638 (Informational), July 1999.

[9] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons not to deploy RED. In *Proceedings of the 7th IFIP Workshop on Quality of Service, London*, Piscataway, NJ, USA, June 1999. IEEE.

[10] Hartmut Ritter. Bedarfsorientierte Dienstgüteunterstützung durch adaptive Endsysteme. PhD Thesis, University of Karlsruhe, Germany, September 2001.

[11] Nabil Seddigh, Biswajit Nandy, and Juha Heinanen. An Assured Rate Per-Domain Behaviour for Differentiated Services. Internet-Draft draft-ietf-diffserv-pdb-ar-01, July 2001.

[12] Stefan Sellschopp. Untersuchung von Dienstgütemechanismen zur differenzierten Behandlung von Paketen verschiedener Prioritätsklassen im Internet der nächsten Generation. Diploma Thesis, Institute of Telematics, University of Karlsruhe, January 2002.

[13] Ralf Steinmetz. *Multimedia-Technologie*. Springer-Verlag. ISBN 3-540-67332-3, 3. edition, 2000.

[14] Klaus Wehrle. Flexible und skalierbare Dienstgütemechanismen für das Internet der nächsten Generation. PhD Thesis, University of Karlsruhe, ISBN 3-8322-0491-1, Shaker Publishing, Aachen, Germany, July 2002.

[15] Klaus Wehrle, Roland Bless, Kathleen Nichols, and Brian Carpenter. A Lower Effort Per-Domain Behavior for Differentiated Services. Internet-Draft, June 2002.

Dr.-Ing. Klaus Wehrle studied Computer Science at the University of Karlsruhe from 1993 till 1999. Afterwards he joined the Institute of Telematics as Research Assistant working on the field of Quality of Service support for the next generation Internet. During his PhD studies he developed the Karlsruhe Implementation architecture of Differentiated Services (KIDS), which is a flexible framework for the construction, implementation and evaluation of individual QoS mechanisms. On this basis he investigated extensively QoS guarantees of Differentiated Services PHBs and PDBs and developed new services (e.g. Quick Forwarding and Limited Effort) respectively. He also introduced parts of his work in the IETF standardization process, e.g. Limited Effort and Multicast issues on Differentiated Services.

Klaus Wehrle finished his PhD at the University of Karlsruhe in July 2002 with honors and will join the International Computer Science Institute (ICSI) at the University of California at Berkeley (UCB) as postdoc in September 2002.

Uwe Walter began his Computer Science study at the University of Karlsruhe in 1996. After his intermediate diploma, he concentrated on various topics in the telematics field. For his practical thesis he implemented and evaluated traffic conditioners in a lab testbed and analyzed forwarding mechanisms for high priority data packets in a simulation model during his master thesis, for which he received his diploma degree in 2001.

Since the beginning of 2002, he works as a research assistant at the Institute of Telematics at the University of Karlsruhe, where he is appointed to the quality of service field in the research group of Prof. Dr. Martina Zitterbart. Currently, he is deployed, in cooperation with several other universities and a responsible partner company, in a research project about next generation networks that deals with different topics, like for example scheduling mechanisms, network admission control, network control and the impact of all these on the provided quality of service.