

A Comparison of Queue Management Algorithms for Intra-Flow Loss Control

Henning Sanneck and Michael Zander

GMD Fokus

Kaiserin-Augusta-Allee 31

D-10589 Berlin, Germany

Abstract—Multimedia flows are inherently inhomogenous, i.e. packets within a flow are of different importance for actual user perception. When transmitting such flows with real-time constraints in networks like the Internet which offer no reliability of transmission, some packet loss is inevitable. The perceptual impact of these losses is then amplified by the arbitrary distribution of packet losses within the flow which impairs the reconstruction of application data units (ADUs) at the receiver, as well as the performance of end-to-end loss recovery mechanisms. To control the loss distribution within a flow (“intra-flow” QoS) typically filtering higher-layer information within the network is proposed, which is both expensive in terms of resources, as well as undesirable with regard to network security. We compare two novel queue management algorithms which improve the intra-flow QoS without higher-layer filtering. The first algorithm is called DiffRED (Differential Random Early Detection). It differentiates between packets marked by the sender as either more or less eligible to be dropped in comparison to unmarked packets without keeping any per-flow state. On the contrary, the other algorithm called PLoP (Predictive Loss Pattern) operates without per-packet marking, yet with keeping partial per-flow state. We introduce simple metrics to describe the loss process of individual flows and present simulation results with voice as foreground traffic using the proposed methods in a multi-hop topology. We find that both algorithms do not have a significant impact on the background traffic. For the given scenario algorithms using packet marking are found to be superior because for the foreground traffic a high probability for short bursts with potentially high perceptual impact can be traded against a higher probability for isolated losses as well as higher (but acceptable) probability for very long loss bursts.

I. INTRODUCTION

Recently, we have seen research efforts on how to use information on a flow’s structure (e.g. the association of packets to frames in an Application Data Unit - ADU) to allow a graceful degradation of the flow when no mechanisms for differentiation between flows are present (“best effort” Internet). We describe these mechanisms with the term “intra-flow” QoS enhancement (as opposed to “inter-flow” QoS where differentiation between flows takes place). For video traffic there have been several proposals (e.g. Frame-Induced Packet Discarding [1], Transcoding, Transform Coefficient Filters), some of which also include an alignment with inter-flow QoS mechanisms [2], [3]. However, these application-level approaches typically suffer from adding significant complexity to nodes interior to the network, contradict with network security constraints and are generally very dependent on the supported payload types which are subject to change over time.

Due to the low per-flow bandwidth for real-time voice, most of the approaches mentioned above for video do not apply. A low-bitrate voice stream typically can neither be source-rate-adaptive nor easily be filtered/transcoded further. Thus, in the absence of inter-flow protection, it needs to be augmented with end-to-end loss recovery. Due to real-time constraints, open-loop error control is used (FEC, [4], [5]), in particular by adding an additional lower quality low bitrate source coding [6]. Alternatively or in combination with FEC, we can exploit long-term correlation within the speech signal for concealment of the signal degradation [7], [8].

However, FEC and loss concealment are limited in the number of consecutive packet losses which can be treated. For FEC, the limitation lies in the additional data and delay overhead necessary to detect and recover consecutive losses. For concealment, the limitation is due to the assumption of quasi-stationarity for speech. This is only valid for a time period typically equivalent to one or two packets. Given these constraints, concealment and forward error recovery approaches become much less efficient as the loss burstiness increases, as shown e.g. in [9], [10].

The previous arguments underline the importance of mapping application requirements with regard to their ADU format and end-to-end quality enhancement capa-

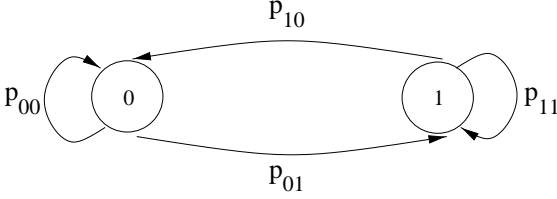


Fig. 1. Gilbert Model

bilities to network mechanisms effectively controlling the *distribution* of losses within a flow.

We compare simple network mechanisms which allows loss control on a per-flow basis which bridge the huge gap between just employing end-to-end loss recovery mechanisms and full deployment of service differentiation/reservation in the network including admission control, negotiation of service level agreements and charging/accounting. We show how applying the algorithms to best-effort voice flows significantly improves application-level QoS without impairing the quality of other flows.

The structure of the paper is as follows: Section II. introduces simple metrics to describe the loss process. In section III. we present the DiffRED [11] algorithm which is an extension to the Random Early Detection (RED, [12]) concept. Section IV. introduces the PLoP algorithm [13]. Then, in section V. we compare both algorithms for a multi-hop scenario by simulation. In section VI. we summarize our findings and conclude the paper.

II. SIMPLE INTRA-FLOW LOSS METRICS

Intra-flow loss metrics¹ introduced up to now (see, e.g., [14], [15]) have been mainly used for admission control, i.e. in the access control path of multiplexers. In contrast, we consider a "best effort" Internet scenario where real-time flows can start and end at any time without explicit setup, i.e. the network has no a-priori knowledge of connections, and thus intra-flow QoS has to be enforced in the data path.

To characterize the behaviour of the network as seen by one flow, we use the well-known Gilbert model (Fig. 1). The system can be completely described by the probability p_{01} for a transition from state 0 (no loss) to state 1 (loss) and the probability p_{11} to remain in state 1. The probability p_{11} represents the *conditional loss probability* clp . The probability of being in state 1, representing the mean loss, is called *unconditional loss probability* ulp and can be computed as follows [16]:

$$ulp = \frac{p_{01}}{1 - p_{11} + p_{01}} \quad (1)$$

¹"Intra-flow" loss probabilities are also referred to as "short-term" loss probabilities because they are typically using random variables describing "close" loss events in terms of the packet sequence.

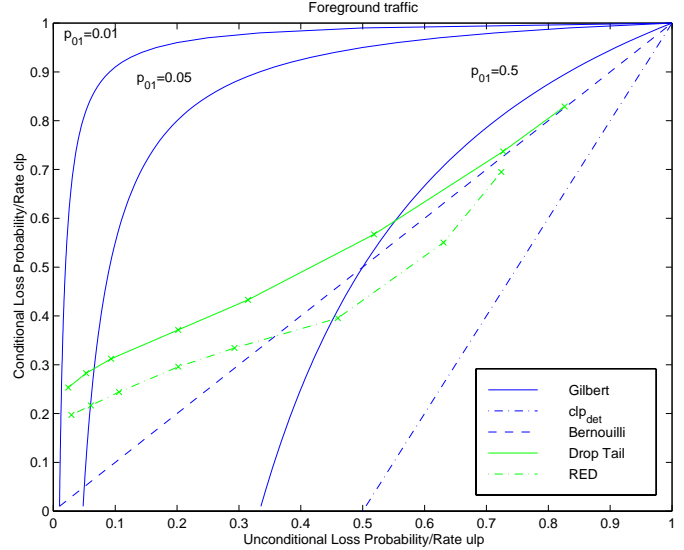


Fig. 2. Conditional Loss Probability vs. Unconditional Loss Probability: Gilbert Model, clp -bound clp_{det} and simulations of Drop-Tail and RED algorithms for foreground traffic

The Gilbert model implies a geometric distribution of the probability for the number of consecutive losses k , $(1 - clp)clp^{k-1}$, which is known to approximate well the head of the loss distribution of actual traces. (The tail of the distribution is typically dominated by few events, caused e.g. by link outages and route flappings, and cannot be captured by a simple model [16], [17].) Fig. 2 shows how the (clp, ulp) space is covered by the Gilbert model using p_{01} as a parameter.

If losses of one flow are correlated (i.e. the loss probability of an arriving packet is influenced by the contribution to the state of the queue by a previous packet of the same flow and/or both the previous and the current packet see bursty arrivals of other traffic, [18]) we have $p_{01} \leq clp$ and thus $ulp \leq clp$ (upper half of Fig. 2). For $p_{01} = clp$ the Gilbert model is equivalent to a 1-state (Bernouilli) model with $ulp = clp$ (no loss correlation).

Fig. 2 leads us to the conclusion that simple queue management algorithms can be designed that allow the adjustment of the conditional loss probability for individual flows², while keeping the unconditional loss probability within a controlled bound around the value that is de-

²Note that by modifying the queue management algorithm, we cannot change the conditional loss probability clp below a theoretical limit. This limit can be explained as follows: clearly the clp can be zero up to $ulp = 0.5$. Then, a deterministic loss pattern with every other packet lost is reached. When further increasing the loss rate, even when considering a deterministic loss pattern, burst losses cannot be avoided. This lower bound is thus given by the deterministic conditional loss probability clp_{det} :

$$clp_{det} = \begin{cases} 0 & 0 \leq ulp < 0.5 \\ 2ulp - 1 & 0.5 \leq ulp \leq 1 \end{cases}$$

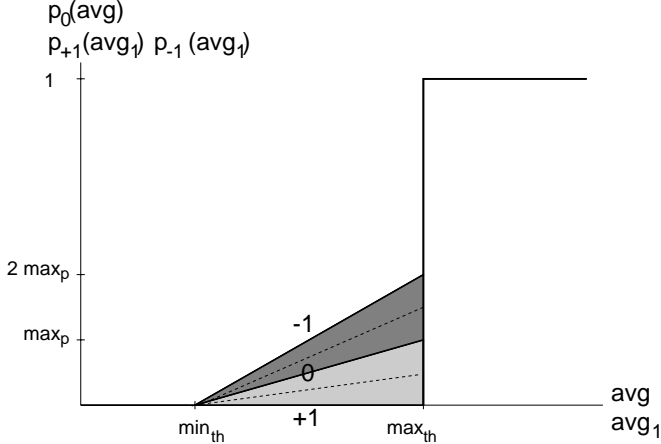


Fig. 3. DiffRED drop probabilities as a function of average queue sizes

terminated by the background traffic load, buffer size, and scheduling policy, but not necessarily by the queue management algorithm itself. In the following we call flows sharing a queue under the control of such an algorithm *foreground* traffic (FT) and the remaining flows in that queue *background* traffic (BT).

With the RED [12] algorithm there exists already a queue management algorithm whose modifications to the queue behaviour can be described with the Gilbert model parameters previously introduced. To be able to accommodate bursts in the queue, as well as not to over-react during transient congestion, the instantaneous queue size is low-pass filtered resulting in an *average queue size* (avg) which is used to compute the drop probability. By employing RED, the parameter p_{01} of the queue is thus increased by gradually increasing the packet drop probability (according to the measured average queue size) before the queue is completely filled. RED was designed to signal congestion to adaptive flows (TCP) and to reduce the average delay independently of the association of packets to flows. However, being interested in the clp , we know that for a given ulp , increasing p_{01} amounts to a reduction in the clp . This effect can be seen in Fig. 2 for simulations we conducted with parameters detailed in the appendix (over a single gateway). For all ulp values, the conditional loss probability when using RED is below that for a Drop Tail queue. Only under heavy overload (when the RED algorithm is also just tail dropping most of the time), the RED curve approaches the Drop Tail one.

III. THE DIFFERENTIAL RED (DIFFRED) ALGORITHM

One approach to realize inter-flow service differentiation using a single queue is RIO ('RED with IN and OUT', [19]). With RIO, two average queue sizes are computed:

one just for the IN packets and another for both IN and OUT packets. Packets marked as OUT are dropped earlier (in terms of the average queue size) than IN packets.

RIO has been designed to decrease the ulp seen by particular flows at the expense of other flows. In this work however, we want to keep the ulp as given by other parameters while modifying the clp parameter for the foreground traffic. Fig. 3 shows the conventional RED drop probability curve (p_0 as a function of the average queue size for all arrivals avg), which is applied to all unmarked ("0") traffic (background traffic: BT). Foreground traffic (FT) packets marked as less eligible for a drop ("+1") are dropped with a probability as given by the lower thick line. This lower probability is compensated by the higher drop probability for the foreground traffic packets marked as ("-1"), i.e. packets more eligible for a drop. This implies that the initial ratio of +1 to -1 packets of a flow must be 1. Therefore in addition to the conventional RED behaviour, the DiffRED implementation should also monitor the +1 and -1 arrival processes [11]. If the ratio of +1 to -1 packets at a gateway is not 1 (either due to misbehaving flows or a significant number of flows which have already experienced loss at earlier hops) the -1 loss probability is decreased and the +1 probability is increased at the same time thus degrading the service for all users³. The shaded areas above and below the $p_0(avg)$ curve (Fig. 3) show the operating area when this correction is added.

In [11] it has been shown that using only the conventional RED average queue size avg for DiffRED operation is not sufficient. This is due to the potentially missing correlation of the computed avg value between consecutive +1 and -1 arrivals, especially when the share of the FT traffic is low. As this might result in a unfair distribution of losses between the FT and BT fractions, a specific avg_1 value is computed by sampling the queue size only at FT arrival instants.

In summary the key differences between DiffRED and RIO are

- "Differential" loss probability curves (a higher loss probability of one packet is compensated by a lower loss probability for another packet)
- Sub-sampling of the queue length value on FT arrival instants to allow for a fair distribution of losses between FT and BT
- Monitoring of the ratio between +1/-1 arrivals to adjust their loss probabilities in case a mismatch (ratio $\neq 1$) between them exists

Thus, a service differentiation for foreground traffic is possible which does not differ from conventional RED behaviour in the long term average (i.e. in the ulp). Ad-

³Another option, yet with significantly higher overhead, would be to identify and deny access to the misbehaving flows [20].

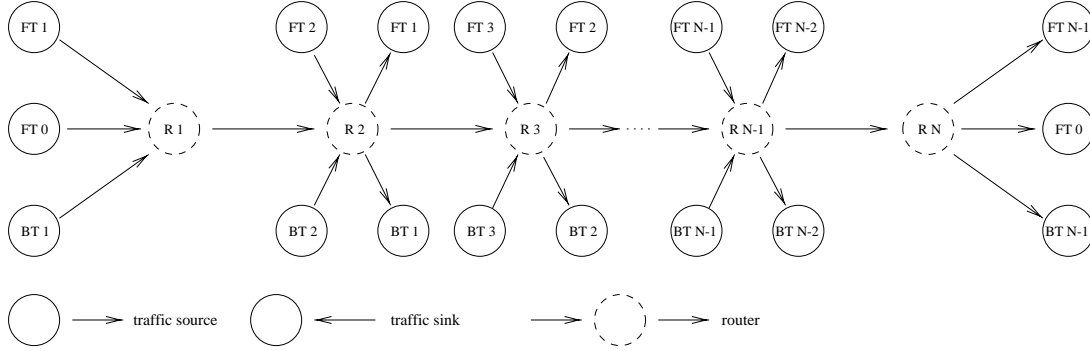


Fig. 4. Topology of simulated network

ditionally, packets of one flow carrying different markers are not reordered.

IV. THE PREDICTIVE LOSS PATTERN (PLOP) ALGORITHM

The PLOP algorithm aims at equally distributing necessary packets drops within a single queue between flows belonging to a certain group of flows with similar properties and QoS requirements (foreground traffic: FT). This is done to minimize violations of the given advance characterization of the flow’s sensitivity to burst losses (“drop profiles”). The task of a “drop profile” is to translate the applications’ end-to-end QoS requirements (i.e. the minimization of the conditional packet loss probability in our case) to a per-packet behaviour of a queue management algorithm at a single node.

For voice traffic we define a simple profile of the conditional drop probability $P_D(x|x-1) = x \bmod 2, x > 0$. $P_D(x|x-1)$ gives the probability used in a *drop experiment* (i.e. a random number is generated and compared against $P_D(x|x-1)$). Note that this profile does not designate consecutive packets (sequence number) of the flow, but packets consecutively subject to a drop experiment. Thus the profile describes rather the worst case, where during times of congestion every packet of a flow is subject to a drop experiment. If this profile is successfully enforced at a node for a flow which previously experienced no losses, the resulting conditional loss probability for that flow at the node is 0.

A. Description of the algorithm

When the queue length exceeds its threshold, a packet is selected to be dropped. After the first drop of a packet of a particular FT flow, the flow ID and the index referring to a corresponding drop probability of the profile for the next drop are recorded in the flow table. The flow ID is the [protocol ID, src addr/port, dst addr/port] tuple for IPv4. With IPv6 the flow label can be used.

When another FT packet should be dropped a drop experiment is performed. The table is checked, whether the ID of the selected packet has already been stored. If true, a random number is generated and the packet is dropped with a probability as found in the table record and the index into the profile within the flow table is updated. If this drop experiment does not result in an actual drop, the packet is marked as a “survivor” and the next packet matching the FT requirement is searched for in the queue (“force drop”, see [13] for the algorithm pseudo code). This procedure is repeated until an actual drop has taken place. If the end of the queue is reached (i.e. no adequate replacement packet for the original packet was found: “force failure”), the original packet is dropped.

V. RESULTS

We employ the same simulation environment as in section II. with the parameters as given in the appendix, however using a network of several hops (Fig. 4 shows the used topology). Foreground traffic consists of several flows which have voice data characteristics (to enable DiffRED operation, every voice source marked its packets alternately with +1 and -1). The foreground traffic share of the offered load $\frac{\lambda_{FT}}{\lambda}$ was set to 10%. Details of the background traffic are described in the appendix. The traffic intensity at every hop is fixed at $\rho = 1.0$. In the following, four algorithms are evaluated (the queue length is 20 packets for all algorithms):

- **Drop Tail (DT)**,
- **Predictive Loss Pattern (PLOP)**,
- **Optimal Predictive Loss Pattern (OPLP)** and
- **Differential RED (DiffRED)**.

The OPLP algorithm works exactly as the PLOP algorithm, however it keeps state about the sequence numbers of packets of a flow seen (see section IV.). OPLP thus

gives a good impression where the performance limit of algorithms working only locally (without inter-hop communication by e.g. packet marking) is, yet this algorithm does not seem viable in real high-speed network environments (due to performance and security constraints).

The performance of the respective algorithms on one isolated hop has been evaluated in [11] and [13]. In this work we will also look at the behaviour along a multi-hop path. The foreground traffic consists of flows which pass through the whole path and are our main focus of interest (FT0) and flows which simulate cross traffic (FT x). The share of cross FT within the FT is 50%. Routers and BT sources and destinations are identified by R x and BT x respectively where x is a number for the path from the source x to the destination x . At every node also new background cross traffic is injected. Flow monitors count all packet arrivals, packet drops, length of loss bursts and observe the sequence numbers of the flows.

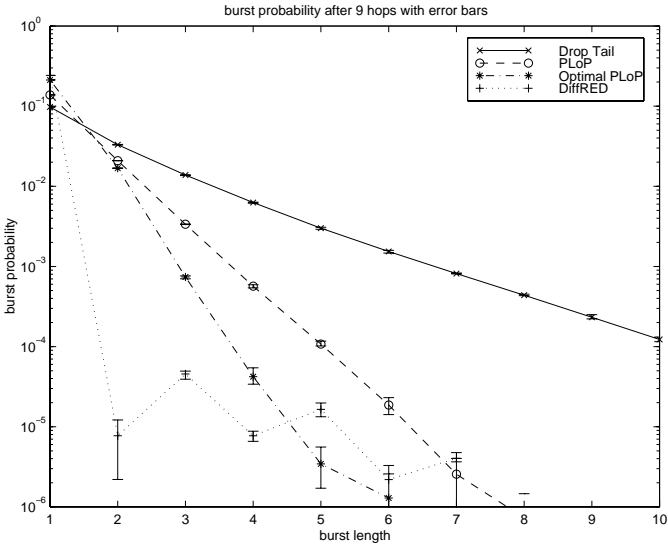


Fig. 5. Burst probability as a function of burst length k after 9 hops

Figure 5 shows the burst probability dependent on the burst length k for a nine-hop topology. The given results are the mean values of all FT0 flows. We also plot error bars giving the standard deviation for the averaged values (this is to verify that every flow of a group has identical behaviour seen over the entire simulation time).

We can observe that DiffRED shapes the burst probability curve in the desired way: most of the probability mass is concentrated at isolated losses ($k = 1$) and all burst probabilities for $k > 1$ are at least three order of magnitude smaller. The other three algorithms show (roughly) only an exponentially decreasing burst loss probability with increasing burst length (with different slopes demonstrating the quality versus state tradeoff). Thus, considering voice as the foreground traffic of interest, with DiffRED a large number of short annoying

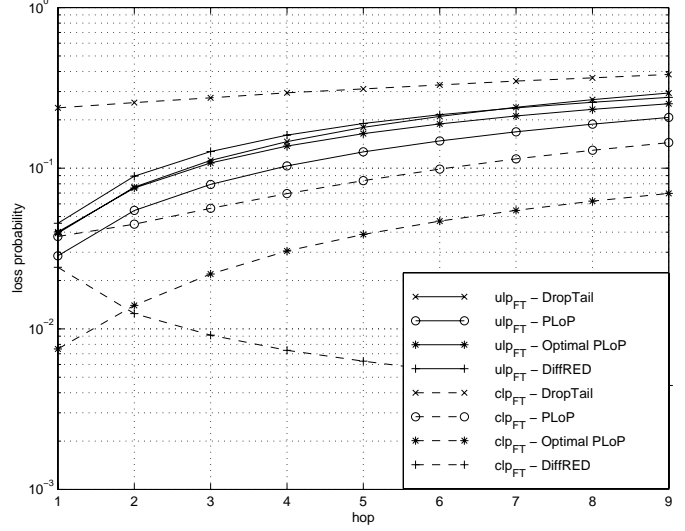


Fig. 6. Development of FT ulp and clp on the transmission path

bursts can be traded against a larger number of isolated losses as well as very long loss bursts. Avoiding longer loss bursts which are perceived as outages [21] is however better achieved by PLoP and OPLP.

For a complete discussion of the loss process influenced by the respective algorithms, we also have to look at the unconditional loss probability (obviously the ulp when using different algorithms needs to be approximately equal to allow a fair comparison between algorithms). The conditional loss probability (clp) then allows us to describe the performance with regard to burst loss in a comprehensive way.

Figure 6 shows how the unconditional loss probability and conditional loss probability of the foreground traffic develop through the path. The ulp values for all algorithms differ only insignificantly as desired for a meaningful comparison of the burst loss properties. However the clp results are very different for every method. On every hop DiffRED can protect "+1" packets by early-dropping "-1" packets using avg_1 as "memory" about the FT flows. OPLP even with keeping individual state on the sequence numbers can only choose among the packets currently present in the queue (the "memory", though being per-flow orientated, is limited to the queue size) and might not find an adequate victim (force failure). The DiffRED algorithm thus provides a path orientated protection through the packet marking: an increasing number of losses along the path increase the ulp but the clp decreases.

To explain this we introduce the occurrence o_k of a burst length k [16]. Thus for a given number of packet arrivals a (experiencing $d = \sum_{k=1}^{\infty} ko_k$ drops) of a flow we have the mean loss rate (ulp for $a \rightarrow \infty$) $p_L = \frac{d}{a}$. With $b = \sum_{k=1}^{\infty} (k-1)o_k$ being the frequency of "two consecutive packets lost", we calculate a conditional loss rate as

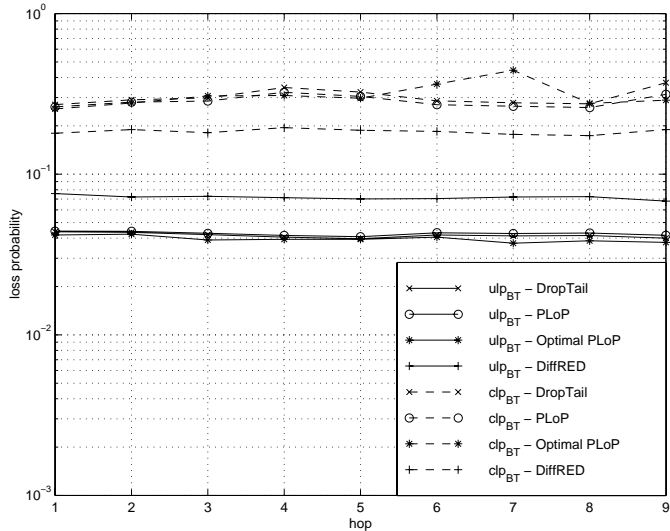


Fig. 7. BT (cross traffic) ulp and clp values at the hops 1-9

$p_{L,cond} = \frac{b}{d}$ (clp for $d \rightarrow \infty$). Using DiffRED at every hop ”+1” packets are dropped with lower probability than ”-1” packets. The occurring losses are concentrated on the ”-1” packets. Thus it happens that the denominator of $p_{L,cond}$ increases faster than the numerator, resulting in a decreasing clp simultaneously to an increasing ulp .

Another interesting issue is the behaviour of the algorithm relative to the background traffic flows, i.e. the fairness to uncontrolled traffic. Figure 7 shows the values for ulp and clp at every hop (we averaged the results for one flow group (H-type BT)). These values are not cumulative values but computed only for one hop because this cross traffic uses only one hop of the path respectively.

The almost identical ulp curve of DT, PLoP and OPLP shows that all three algorithms have only minor influence on the background traffic. The DiffRED algorithm retrieves some of its performance from the BT but at a tolerable level.

VI. CONCLUSIONS

We have introduced simple metrics to describe the loss process of individual flows and presented simulation results in a multi-hop topology for a voice service using queue management algorithms for burst loss control. Two classes of algorithms have been compared: the first class uses packet marking to designate drop preference without keeping local per-flow state. The second class on the contrary operates purely local (the sender or other nodes are not involved in the scheme), but keeps state for the protected flows. We find that both types of algorithms do not have a significant impact on conventional traffic. It is possible to control the loss characteristics of individual flows while keeping their unconditional loss probability within a controlled bound around the value expected using con-

ventional Drop Tail or RED algorithms. For the given scenario algorithms using packet marking are found to be superior because a high probability for short bursts with potentially high perceptual impact can be traded against a higher probability for isolated losses as well as higher (but acceptable) probability for very long loss bursts.

APPENDIX

We implemented the PLoP, OPLP and DiffRED algorithms into a modified version of the NS-2 network simulator [22], which allows tracing of the occurrence of burst losses for individual flows. In our simulation the bottleneck links (i.e. the capacity of the routers’ egress interface) have a link-level bandwidth of $\mu = 1920kbit/s$. Flows are fed to the routers over $10Mbit/s$ links.

We used the same traffic model as in [11] and [13], that reflects results from various recent Internet Access-LAN and Internet backbone measurements [23]–[25]: the majority of traffic (in terms of flows and volume) are http transfers (“H-type” background traffic). The rest are mostly short-lived flows dominated by DNS traffic (“D-type” background traffic), which has a relatively large share of the active flows, yet only a small share of the traffic volume⁴. The values we chose for modeling of individual sources are shown in Table I. To model Web traffic we use a Pareto distribution [26] both for the ON and OFF periods of the source. By using a variance-time ($var(X(m)) - m$) plot [27], describing the variance of the process of arrivals X dependent on the scale of averaging m , we determined that the aggregation of the described background traffic sources produces long-range dependent traffic. As the DiffRED algorithm tries to influence the loss burstiness of individual flows, it is crucial to reflect the existing “burstiness on all time scales” of the aggregate arrival process in the model. To model voice sources with silence detection, we employed a model widely used in the literature (see e.g. [15]) where ON (talkspurt) and OFF periods are exponentially distributed with a speaker activity of 36%. Every voice source marks its stream with a +1, -1, +1, -1, ... profile.

Table I also gives “raw” peak bandwidth and packet sizes (i.e. including packet header overhead⁵). Packet inter-departure times within a burst are uniformly distributed in the interval $[0.95I, 1.05I]$ (with I being the packet inter-departure time calculated from the values of Table I) to avoid phase effects caused by the exact timing of packet arrivals in the simulator.

We found a simulation time of $5 \cdot 10^4$ seconds (13.9 hrs.⁶ with the number of packet arrivals ranging from $16 \cdot 10^6$

⁴The small per-flow bandwidth of the D-type BT allows us to set the background traffic load with a relatively fine granularity.

⁵We assume 8 bytes link level overhead and 20, 20, 8, 12 bytes IP-, TCP-, UDP-, RTP-packet overhead respectively.

⁶The initial 10^4 s were discarded from the datasets.

TABLE I
Source model parameters

Traffic type	H-BT	D-BT	FT (voice)
flow share (%) of BT	75	25	-
peak bandwidth ($\frac{kbit}{s}$)	256	32	83.2
packet size (bytes)	560	128	208
on/off distribution	Pareto	Expo.	Expo.
shape parameter	1.9	-	-
mean burst (packets)	20	4	18
mean ontime (s)	0.35	0.13	0.36
mean offtime (s)	0.7	0.13	0.64

to $27 \cdot 10^6$) sufficient for the Pareto sources to "warm up" and thus to guarantee that the traffic shows long-range dependence as well as to result in a statistically relevant number of drop events even for low loss rates as a basis for performance measures ($p_{L,cond}$).

The (Diff)RED parameters used for all simulations are as follows: $min_{th} = 5$, $max_{th} = 15$, $max_p = 0.1$ and $w_q = w_{q,1} = 0.002$ [11].

REFERENCES

- [1] S. Ramanathan, P.V. Rangan, and H. Vin, "Frame-induced packet discarding: An efficient strategy for video networking," in *Proceedings of NOSSDAV '93*, 1993, pp. 173–184.
- [2] L. Wolf, R. Herrtwich, and L. Delgrossi, "Filtering multimedia data in reservation-based internetworks," Technical Report 43.9608, IBM European Networking Center, August 1994.
- [3] R. Wittmann and M. Zitterbart, "AMnet: Active multicasting network," in *Proc. of Intern. Conf. on Communications (ICC'98)*, Atlanta, GA, June 1998.
- [4] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, Sept./Oct. 1998.
- [5] C. Perkins and O. Hodson, "Options for the repair of streaming media," RFC 2354, IETF, June 1998, <ftp://ftp.ietf.org/rfc/rfc2354.txt>.
- [6] C. Perkins, I. Kouvelas, O. Hodson, M. Handley, and J. Bolot, "RTP payload for redundant audio data," RFC 2198, IETF, September 1997, <ftp://ftp.ietf.org/rfc/rfc2198.txt>.
- [7] H. Sanneck, A. Stenger, K. Ben Younes, and B. Girod, "A new technique for audio packet loss concealment," in *Proceedings IEEE Global Internet 1996 (Jon Crowcroft and Henning Schulzrinne, eds.)*, London, England, November 1996, pp. 48–52, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9611:-New.ps.gz>.
- [8] H. Sanneck, "Concealment of lost speech packets using adaptive packetization," in *Proceedings IEEE Multimedia Systems*, Austin, TX, June 1998, pp. 140–149, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9806:-Adaptive.ps.gz>.
- [9] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," in *Proceedings ACM SIGCOMM '90*, San Francisco, CA, June 1990, pp. 124–131.
- [10] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high-speed networks," *IEEE Transactions on Information Theory*, vol. 39, pp. 98–108, January 1993.
- [11] H. Sanneck and G. Carle, "A queue management algorithm for intra-flow service differentiation in the "best effort" Internet," in *Proceedings of the Eighth Conference on Computer Communications and Networks (ICCCN '99)*, Natick, MA, October 1999, pp. 419–426, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9910:Intra-Flow.ps.gz>.
- [12] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.
- [13] H. Sanneck and G. Carle, "Predictive loss pattern queue management for Internet routers," in *Internet Routing and Quality of Service*, S. Civanlar, P. Doolan, J. Luciani, R. Onvural, Editors, *Proceedings SPIE Vol.3529A*, Boston, MA, November 1998, pp. 205–216, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9811:-Predictive.ps.gz>.
- [14] T. Miyata, H. Fukuda, and S. Ono, "New network QoS measures for FEC-based audio applications on the Internet," in *Proceedings IEEE IPCCC 1998*, Tempe/Phoenix, AZ, USA, February 1998, pp. 355–362.
- [15] R. Nagarajan, J. Kurose, and D. Towsley, "Finite-horizon statistical Quality-of-Service measures for high speed networks," *J. High Speed Networks*, December 1994.
- [16] H. Sanneck and G. Carle, "A framework model for packet loss metrics based on loss runlengths," in *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000)*, San Jose, CA, January 2000, pp. 177–187, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0001:Run-length-Metrics.ps.gz>.
- [17] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for interactive audio in the Internet," in *Proceedings IEEE Infocom '99*, New York, NY, March 1999.
- [18] H. Schulzrinne, J. Kurose, and D. Towsley, "Loss correlation for queues with bursty input streams," in *Proceedings ICC '92*, Chicago, IL, 1992, pp. 219–224.
- [19] D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," Technical Report, MIT LCS, 1997, <http://diffserv.lcs.mit.edu/Papers/exp-alloc-ddc-wf.pdf>.
- [20] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," Technical Report, Network Research Group, LBNL, February 1997.
- [21] J. Gruber and L. Strawczyynski, "Subjective effects of variable delay and speech clipping in dynamically managed voice systems," *IEEE Transactions on Communications*, vol. Vol. COM-33, no. 8, August 1985.
- [22] UCB/LBNL/VINT, *Network simulator ns-2*, October 1998, <http://www-mash.cs.berkeley.edu/ns/ns.html>.
- [23] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings INET '98*, Geneva, Switzerland, July 1998, http://www.isoc.org/inet98/proceedings/6g/6g_3.htm.
- [24] P. Newman, T. Lyon, and G. Minshall, "Flow labelled IP: Connectionless ATM under IP," in *Proceedings of Network + Interop*, Las Vegas, NV, USA, April 1996.
- [25] M. Iivesmäki, K. Kilki, and M. Luoma, "Packets or ports - the decisions of IP switching," in *Broadband Networking Technologies, Seyhan Civanlar, Indra Widjaja, Editors, Proceedings SPIE Vol.3233*, Dallas, TX, November 1997, pp. 53–64.
- [26] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. Vol. 5, no. 6, pp. 835–846, December 1997.
- [27] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic," in *Proceedings ACM SIGCOMM*, San Francisco, CA, September 1993.