

Predictive Loss Pattern Queue Management for Internet Routers*

H. Sanneck and G. Carle

GMD Fokus

Kaiserin-Augusta-Allee 31 D-10589 Berlin Germany

ABSTRACT

Widespread availability of real-time services is the next challenge for the Internet after the introduction of the WWW has already changed Internet traffic patterns once. As the Internet provides a “best effort” datagram service only, no assurance for actual packet delivery for real-time flows can be given. Most real-time applications exhibit tolerance against occasional loss of packets, but are sensitive to losses that occur in bursts. This is especially significant for a voice service, which we consider as our primary target application in this paper due to its importance in the future Internet, its relatively well-known subjective properties in the presence of packet loss and its simple flow structure. Currently all *hop-by-hop* approaches to enhance the Quality-of-Service for real-time flows either use strict per-flow setup and state maintenance of reservations (Integrated Services) or rely on the sender/ingress router which is unaware of the amount and the location of congestion in the network to mark packets for preferred treatment (Differentiated Services). This results in either high resource consumption in the network (due to a conservative characterization of the application’s requirements) or dissatisfactory perceived quality (due to the toleration of too many burst losses in the network). For interactive voice, *end-to-end* adaptation to the current network load is also difficult to apply, considering the per-flow overhead and usual traffic properties (low bitrate).

We propose an active queue management algorithm called PLoP (Predictive Loss Pattern), that instead of enforcing static reservations tries to give preferred service over short time intervals to particular flows that have previously been discriminated (i.e. lost packets). The probability of a packet drop is made dependent on flow history (previous drops) and flow type. This allows a basic protection of burst-loss-sensitive flows during transient and persistent congestion, keeping only partial per-flow state. The proposed algorithm does not require any specific cooperation of the applications and gives an incentive to build loss-resilient applications using end-to-end QoS enhancement (FEC and/or error concealment), which can build on the enforced predictive loss pattern.

Simulation results show the performance at a congested network element in terms of conditional and unconditional loss probability and processing/state overhead. It is shown that burst loss protection in the data path for periodic foreground traffic (voice) is feasible for a wide range of load conditions. Any impact on non-real-time traffic in terms of the conditional and unconditional loss probability is avoided, as long as the link-speed equivalent buffer is larger than the maximum expected traffic period.

Keywords: packet loss, predictive loss pattern, short-term QoS, queue management, Internet voice transmission

1. INTRODUCTION

Loss and delay tolerance of applications can be described by utility functions.¹ For multimedia applications, utility functions relate the available network resources for a particular flow to the perceived quality and thus to user satisfaction.² Fig. 1 shows a schematic utility curve (A) for interactive voice dependent on the *unconditional loss probability* (*ulp*) (the curves are based on subjective test results of (3-8)). The strong performance degradation starting at low loss rates is due to the missing ability of the source to adapt its rate for delivering a “complete” stream with gracefully degraded quality. Instead the source keeps its rate and lets the network degrade the quality in an uncontrolled manner. This problem can be overcome by either a rate-adaptive⁹ sender or layered transmission

*To appear in Internet Routing and Quality of Service, Proceedings SPIE Vol.3529A

Other author information:

H.S.: Email: sanneck@fokus.gmd.de; Telephone: +49-30-3463-7175; Fax: +49-30-3463-8175.

G.C.: Email: carle@fokus.gmd.de; Telephone: +49-30-3463-7149; Fax: +49-30-3463-8149.

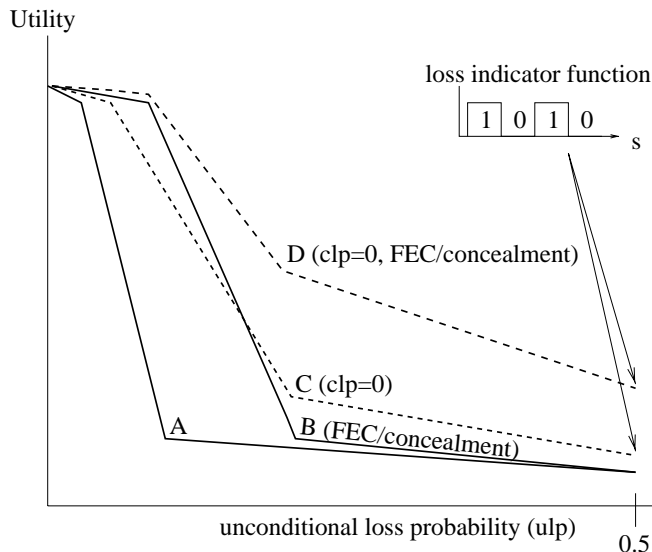


Figure 1. Utility for interactive voice (schematic)

with adaptive receivers¹⁰ and mechanisms assuring fairness in the network and end-systems (otherwise aggressive applications may monopolize the bandwidth). As this is difficult with the currently standardized codecs, another option is to add FEC information¹¹ (e.g. a lower quality low bitrate source coding⁵) to recover a packet or to exploit long-term correlation within the speech signal by concealing^{6–8} the signal degradation to the user. However, as it can be seen from curve (B) for higher loss rates, this curve quickly approaches curve (A). This is due to the increasing correlation of losses (and thus decreasing effectiveness of FEC/concealment). Subjective tests³ showed that it is generally preferable for the resulting speech quality to have a high number of small length gaps ($\approx 20ms$) rather than infrequent occurrence of long gaps (which leads to the loss of entire logical speech elements in the signal). This is reflected in curve (C) using an artificial loss *pattern* of only *singleton* losses, i.e. the *conditional loss probability* is $clp = 0$. In combination, network mechanisms influencing the correlation of losses for individual flows would be able to maintain the gain of FEC¹²/concealment mechanisms also for higher loss rates (curve (D)^{7,8}). In general, probabilistic assumptions about the expected loss pattern can greatly enhance the performance of application level end-to-end loss recovery mechanisms.

Fig. 1 also shows the importance of parameters like clp which describe the *short-term* requirements of applications, reflecting another axis of fairness between different applications, as bursty background traffic can cause burst losses (dropouts) for a voice flow without affecting a long-term fair bandwidth share. Previously, short-term QoS measures^{13–16} were mainly used for admission control (i.e. in the access control path) e.g. for voice multiplexers.^{17,18} In contrast we consider a dynamic scenario (real-time flows can start/end at any time without explicit setup, i.e. no a-priori knowledge of connections) where short-term QoS has to be enforced in the *data path*.

The emerging Differentiated Services (DS) architecture¹⁹ introduces network mechanisms that may influence the correlation of losses. It allows for the specification of preferred treatment of flows on a *per-packet* basis (marking), thus enabling the enforcement of short-term QoS loss patterns. However we need to rely on a co-operating sender (or the first-hop router) for the marking of packets. Additionally, separate treatment of packets belonging to the same flow (in an uncongested state) can lead to reordering. Our view is that voice will be a basic service of the future Internet with usage characteristics like in today's PSTN. Therefore a scalable, pre-configured, open service between best-effort and DS should be realized, i.e. the application chooses the service just by characterizing the packet payload (e.g. with RTP). The type of the network voice service provided can be assumed to change very slowly (if at all) over time (e.g. due to further developments in coding technology). Charging for such a basic service between ISPs could be based on coarse granularities like overall voice traffic volume at the ISP domain boundaries rather than expensive per-flow accounting.

The structure of the paper is as follows: Section 2 explains assumptions and options concerning the design of a burst loss control algorithm. Section 3 presents our proposed algorithm called “Predictive Loss Pattern“ **PLoP**. In section 4 we evaluate a preliminary implementation of the algorithm by simulation.

2. DESIGN OF A BURST LOSS CONTROL ALGORITHM

Looking at a deterministic flow loss pattern of “every second consecutive packet lost” (Fig. 1), we observe that the unconditional loss probability (ulp_{det}) is 0.5 whereas the conditional loss probability (clp_{det}) is 0. Therefore it is reasonable to assume that a simple burst loss control algorithm can be designed, enforcing a clp for a flow which is close to 0, while operating at a $ulp \ll ulp_{det}$ given by parameters (background traffic intensity, buffer size) not controlled by the algorithm. We designate the fraction of flows at a gateway under the control of such an algorithm by *foreground* traffic (FT) and the remaining flows as *background* traffic (BT).

The main goal is to approximate the given loss requirements of the foreground traffic (voice: spread inevitable loss over a larger time period) while at the same time avoiding a negative impact on the background traffic, especially on adaptive BT like TCP or rate-adaptive real-time flows. Unnecessary burst losses should be avoided where possible. “Unnecessary” here means that the impact of a dropped packet on a particular FT flow is much higher than on another previously unharmed one also currently present in the queue. Additionally, the incurred overhead (control state and additional processing) at the gateway has to be adequate for the only partial QoS assurance given.

As we consider voice as our primary target application, we can collect some properties of this foreground traffic which partially ease the design:

1. FT flows do not exhibit significant burstiness compared to the BT. They have comparable packet interarrival times and mean rates (i.e. the mean occupancy of the aggregated FT flows in the queue can be assumed to change slowly).
2. FT does not consume a major fraction of the entire gateway bandwidth (otherwise unfairness to the BT cannot be avoided). This depends on the future ratio between voice and data traffic, but we argue that it safely can be assumed to hold: the per-flow bandwidth decreases due to advances in coding technology. The overall usage of interactive voice will increase slowly if at all and the overall non-voice traffic volume will surely increase.
3. The FT fraction is constituted of several independent flows (otherwise an equal distribution of losses within the FT fraction will be difficult to achieve).

Three **options** for the basic structure of a burst loss control algorithm can be identified:

1. per FT flow queueing (n queues)
2. per FT/BT queueing (2 queues)
3. single queue

We explore only item 3. as it has the desirable feature that the algorithm needs to be active only during times of congestion as well as simplicity, scalability (no scheduling between queues, only queue management is needed) and easy deployment. An example for this third category is RED,²⁰ which influences the probability of a packet drop *without keeping per flow state*. The measurement of an average queue size triggers random suppression of packets with an increasing probability as the average queue size increases. This signals congestion to adaptive flows (TCP), reduces the average delay and allows bursty traffic to be accommodated. It is proposed in ⁽²¹⁾ to extend RED by identifying (and discriminating) misbehaving flows. Instead our solution temporarily protects properly behaving flows (these are however flows which cannot be rate-adaptive), keeping *partial* per-flow state. Guerin et al.²² show, using buffer management with *full* per-flow state, how rate guarantees can be established for individual flows.

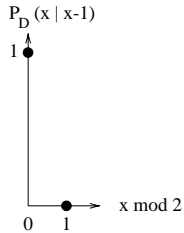


Figure 2. Voice drop profile

3. 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/ 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”).

3.1. 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) to a *per-packet* behaviour of a queue management algorithm at a single node.

A comparable approach is taken by Koodli and Krishna²³ defining an end-to-end “noticeable loss rate” metric, where the application specifies an acceptable task loss of a scheduler over a time window. These requirements are translated to a per-subtask control algorithm at a node. Seal and Singh²⁴ introduce “loss profiles” as pre-defined discarding functions (“clustered”/”random” loss) operating over certain time windows on logical data segments designated by the application. The profiles are then enforced at the transport layer of the source host or an intermediate node.²⁵

For voice traffic we define a simple profile of the conditional *drop* probability $P_D(x|x-1), x > 0$ as in Fig. 2. $P_D(x|x-1)$ 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 s) of the flow, but packets consecutively subject to a drop experiment (index x). 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, the resulting conditional loss probability of a particular flow at this node is 0. This profile does not give information about an actual unconditional loss probability that can be expected, however it clearly establishes an upper bound on the unconditional loss probability $\hat{p}_L = 0.5$.

The same profile as in Fig. 2 could be applied meaningfully to video traffic at the frame level, for coding schemes where every frame has the same importance (e.g. M-JPEG). Then, a technique like Frame-Induced Packet Discarding (FIPD²⁶) can be used for actual enforcement of the profile at the packet level.

The *distribution* of drop profiles can range from hard-coding within the PLoP algorithm (which we assume to be sufficient for a basic voice service, see section 1) up to “active” (per-flow) setup. Details on the distribution are beyond the scope of the paper.

3.2. 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

[†]In our current implementation, the threshold is set to the maximum queue size. However to better accomodate transient congestion and the additional processing time needed to execute the algorithm it seems promising to combine PLoP e.g. with RED²⁰ to control the average queue size. Additionally, the drop probabilities could be weighted with the average queue size.

[‡]Usually the first profile probability is 1 as we only aim to control the *burst* loss characteristics.

```

PLoP()
if queue threshold exceeded
    delete timer
    if (packet ∈ FT) // flow type filter
        status = drop_experiment()
        if (status == FAILED) //“force failure”
            drop // other policy: drop BT packet
        else
            drop
    elseif (not idle)
        if (timer expired)
            delete flow table, go idle
        elseif (timer not running)
            start timer

drop_experiment()
if (flow not in flow table) // flow ID filter
    create flow table entry
    generate random number  $R \in [0, 1]$ 
    if  $R \leq P_D(x|x-1)$  and (packet not “survivor”)
        drop
        return OK
    else // “force drop” of an FT packet
        mark as “survivor”
        if (end_of_queue)
            return FAILED
        else
            lookup next FT packet in queue
            status = drop_experiment()
            return status

```

Figure 3. Predictive Loss Pattern algorithm pseudo code

the next drop is 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. Note that a flow might also consist of aggregated²⁷ “sub”-flows.

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 Fig. 3 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*”), either the original packet or a BT packet is dropped.

3.3. Properties

As enforcing drop profiles also results in establishing an upper bound on the unconditional loss probability (cf. section 3.1), the amount of flows concurrently under PLoP protection has to be limited accordingly. For voice traffic, the maximum flow table size is set to $\lfloor \frac{B\hat{p}_L}{r\alpha} \rfloor$ (with B : gateway bandwidth, $0 < \hat{p}_L < 1$: upper bound on the mean loss rate as determined from the profile, r : rate expected of an individual flow during talkspurts and $\alpha = 0.6$ (conservative) estimate of the speaker activity).

Flow table management policy

Considering a limited flow table size, a flow table management policy defining when a flow ID is added/dropped from the flow table is needed. Two basic flow table management policies can be identified: preemptive and non-preemptive. In the *preemptive* policy the table size is limited and handled in a FIFO way, i.e. if the length of the table is exceeded by adding a new entry, the oldest entry is dropped. The flow table is deleted entirely, when an “uncongested” state[§] persists to avoid keeping old state in the table.

Using a *non-preemptive* policy, all packets belonging to flows not present in the (full) flow table are dropped, because otherwise the minimal guarantee on the loss rate would be violated. Rather than degrading the service given

[§]The “uncongested” state is determined by monitoring the (non-)access to the flow table over a time interval. Note that after expiration of the timer, PLoP stays idle and does not consume any resources.

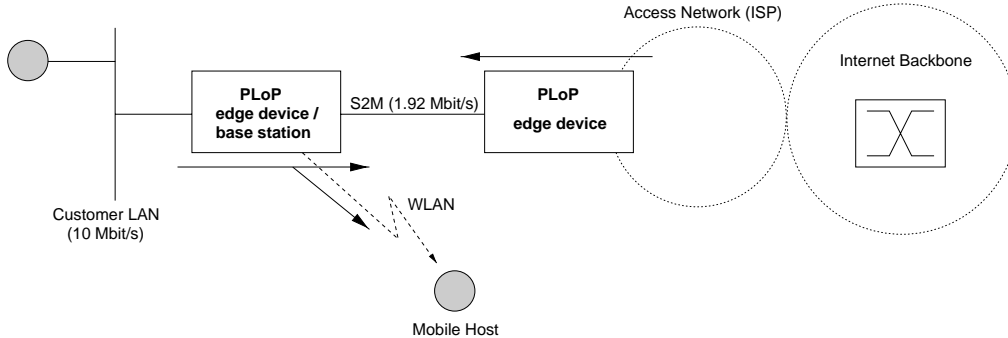


Figure 4. PLoP deployment scenario

to all other flows below the acceptable minimum level, other “calls” are “blocked”. For this policy, additional per flow table entry timers are needed, otherwise entries of inactive flows could persist during congestion. Due to this additional overhead, the preemptive policy is used in our preliminary implementation.

Force failure policy

As the policy for the case when no adequate replacement packet was found in the queue (“force failure”), we adopted dropping of the packet that originally would have been protected. One might argue that a “force failure” is mainly due to a flow occupying more than its fair share of the buffer space which therefore should be discriminated. However without further knowledge (state) of misbehaving flows,²¹ it should be avoided to randomly drop any background traffic. Results of section 4.2 show that (presuming sufficient buffer space) FT flows can be sufficiently protected even under overload conditions. Another reason for a “force failure” can be that only very few FT flows are active at a gateway. Here we argue that the impact of dropping background traffic due to the small overall FT bandwidth is minimal. However, as long as the link-speed equivalent buffer is larger than the FT interarrival time, this type of “force failure” virtually does not occur.²⁸

Choice of the dropping discipline and search direction

The distribution of loss bursts has been shown to be similar for front and tail dropping disciplines^(29,30). Combined with the possibility of searching for PLoP replacement packets from either end of the queue, four strategies exist which lead to marking packets at different queue locations.

All solutions except *drop from front/search from front* lead to accumulation of “survivor” packets in the queue (packets not dropped due to the PLP drop logic should be as close to the head of the queue as possible to avoid unsuccessful drop experiments). See⁽²⁸⁾ for a detailed evaluation.

4. EVALUATION

To assess the performance of PLoP, we evaluated a scenario where several flows experience a bottleneck link (e.g. an small bandwidth access link connecting a customer LAN to an ISP or a base station connecting mobile hosts to a LAN, Fig. 4). In our simulation the bottleneck link has a link-level bandwidth of $\mu = 1920kBit/s$ (ISDN[¶] S_{2M} PRI). Several flows fed to the gateway over $10Mbit/s$ links are multiplexed to either a Drop Tail (DT) or a PLoP output queue.

We implemented the algorithm into a modified version of the NS-2 simulator,³¹ which allows tracing of the occurrence o_k of burst losses of length k for individual flows. 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 occurrence measure of “two consecutive packets lost”, we calculate a *conditional loss rate* as $p_{L,cond} = \frac{b}{d}$ (*clp* for $d \rightarrow \infty$). Note that for longer drop profiles (section 3.1) additional measures are needed^(14-17,28). Additionally, we monitor various PLoP queue parameters.

[¶]We neglect connection setup times and the fragmentation into channels.

	<i>H-type BT</i>	<i>D-type BT</i>	<i>FT (voice)</i>
flow share (%) (of background traffic)	75	25	-
peak bandwidth ($\frac{kBit}{s}$)	256	30...34	83.2
packet size (bytes)	8+20+20+512	8+20+8+92	8+20+8+12+160
on/off distribution	Pareto	Exponential	Exponential
shape parameter	1.9	—	—
mean burst length (packets)	20	4	18
mean ontime (s)	0.35	0.12...0.14	0.36
mean offtime (s)	0.7	0.12...0.14	0.64

Table 1. Source model parameters

4.1. Traffic Model

We use a traffic model, that reflects results from various recent Internet Access-LAN and Internet backbone measurements (e.g.³² and^{33,34}): 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 volume. The values we chose for modeling of individual sources are shown in Table 1. To model Web traffic we use a Pareto distribution³⁵ both for the ON and OFF periods of the source. By using a variance-time ($var(X(m)) - m$) plot,³⁶ 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.²⁸ As the PLoP 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. (17)) where ON (talkspurt) and OFF periods are exponentially distributed with a speaker activity of 36%.

Table 1 also gives “raw” peak bandwidth and packet sizes (i.e. including packet header overhead^{||}). The range of 30...34 $\frac{kBit}{s}$ D-type BT bandwidth and 0.12...0.14s for the on-/offtimes is due to the changing number of flows and load in the experiments which figure below. 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 1) to avoid phase effects caused by the exact timing of packet arrivals in the simulator.

We found a simulation time of 5×10^4 seconds (13.9 hrs.**with the number of packet arrivals ranging from 16×10^6 to 27×10^6) 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 ($p_{L,cond}$). We averaged the results for one flow group (H, D, voice). On the figures 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).

4.2. Results

We set the share of voice traffic to 10% of the gateway bandwidth for all experiments, resulting in six active voice flows. The share of BT traffic (at a traffic intensity $\rho = \frac{\lambda}{\mu} = 1$, λ being the offered load) is set to 80% (18 flows) for H- and 10% (6 flows) for D-type BT respectively. For other traffic intensities, the BT share is varied while keeping the ratio of H- and D-type BT approximately equal, resulting in 12 H-type, 4 D-type ($\rho = 0.66$), up to 24 H-type and 8 D-type flows ($\rho = 1.23$) active.

^{||}We have 8 bytes link level (HDLC) overhead and 20, 20, 8, 12 IP-, TCP-, UDP-, RTP-packet overhead respectively.

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

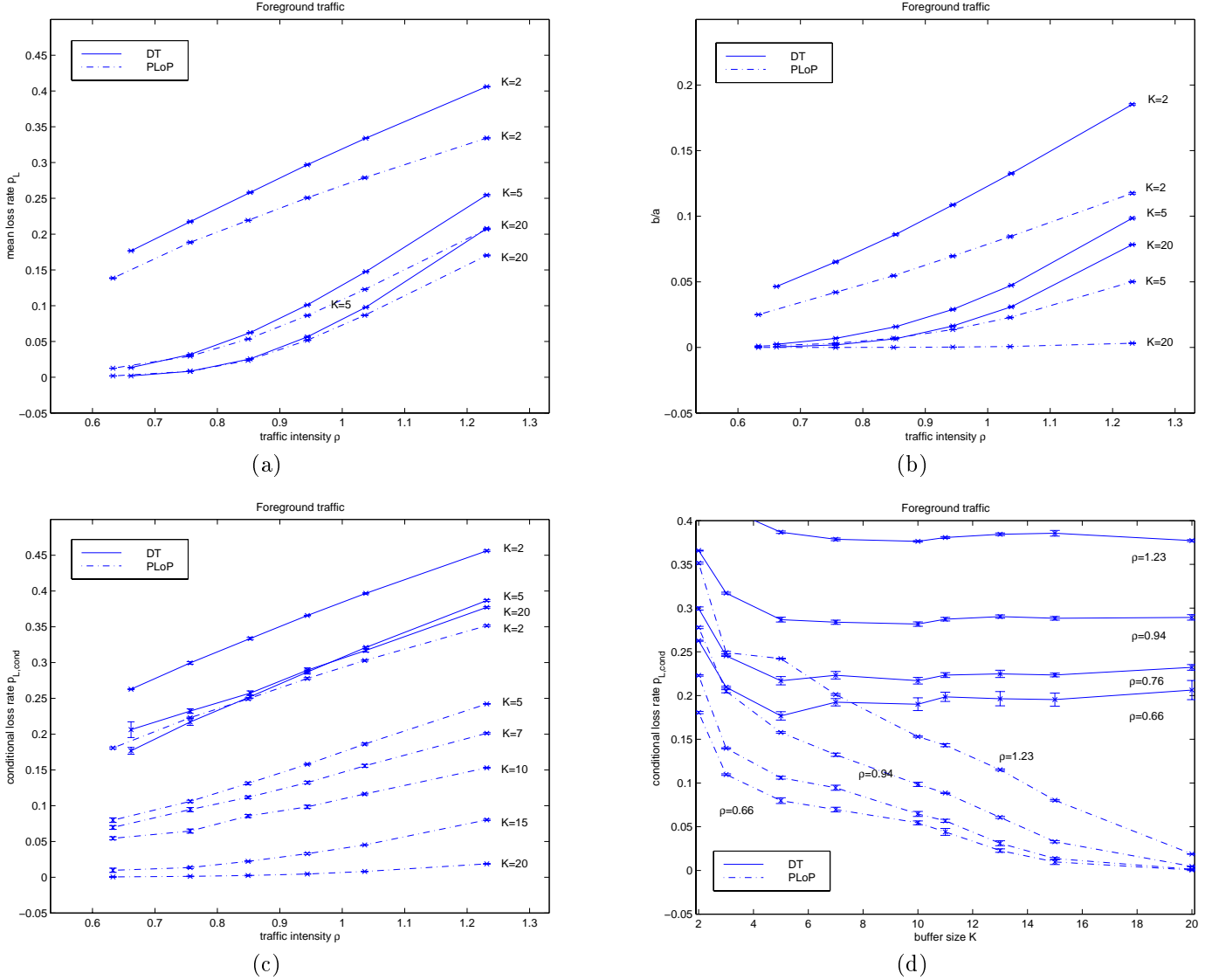


Figure 5. FT performance measures

Fig. 5 (a) shows the mean loss rate p_L as a function of the traffic intensity ρ . Except for low buffer sizes ($K < 5$), we see that for $\rho < 0.9$, p_L has approximately the same value for Drop Tail (DT) and PLoP and thus seems to be acceptable in terms of fairness towards the BT. For higher loads, curves for the PLoP algorithm start to approach their asymptote (maximum possible loss rate) which is given by $\hat{p}_L = 0.5$ ($\hat{p}_L \rightarrow 1$ for DT, section 3.1).

Looking at the conditional loss rate $p_{L,cond}$ in Fig. 5 (c) and (d), we see that for DT, increasing the buffer size (except for very low buffer sizes) has virtually no effect on $p_{L,cond}$. For lower loads $\rho < 0.9$, $p_{L,cond} = \frac{b}{a}$ for $K = 20$ is even larger than $p_{L,cond}$ for $K = 5$. This is *not* due to a larger number of burst losses b for the larger buffer size, as can be seen from Fig. 5 (b) where $\frac{b}{a}$ is decreasing with larger buffer sizes. However the b values, as well as the difference of b for different buffer sizes are small compared to a . Thus, for lower loads (where the queue can drain between bursts) the loss process is dominated by burst losses caused by very large arrival bursts (burst size \gg buffer size K) and singleton losses (which appear only in the d -term of $p_{L,cond}$). Note that the Pareto distribution is heavy-tailed, i.e. significant parts of the probability mass are concentrated at rare (but large) bursts and frequent bursts of only few packets.

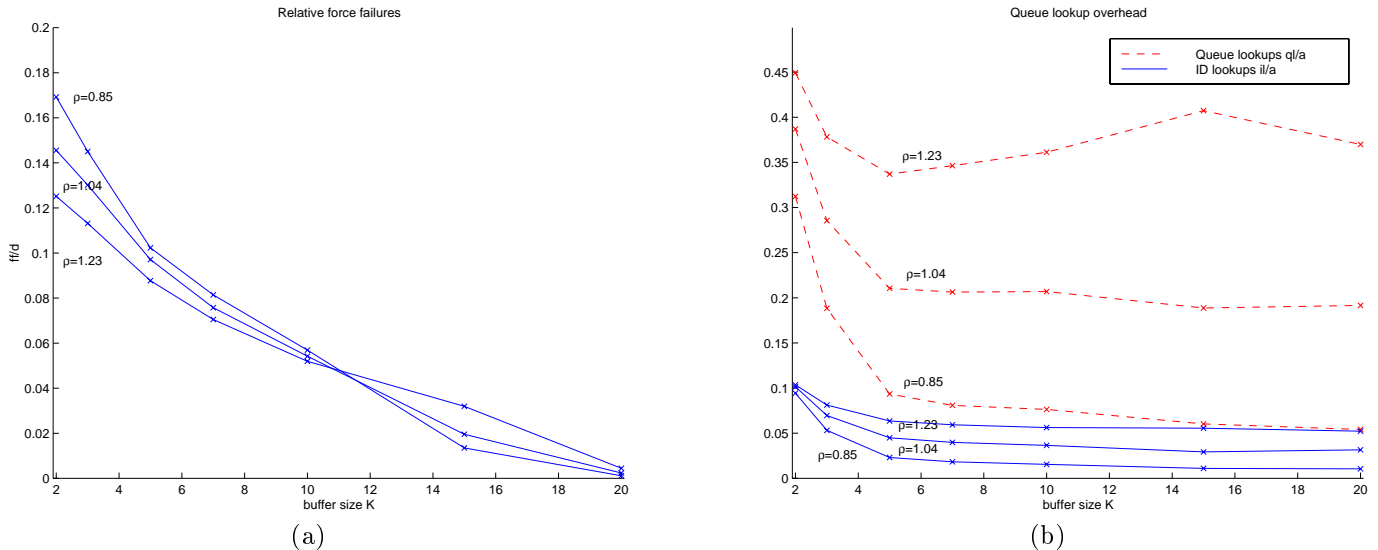


Figure 6. PLoP queue performance parameters

The behaviour of $p_{L,cond}$ for the PLoP algorithm shows that PLoP can exploit larger buffer spaces to avoid burst losses within one flow. Starting from an enhancement of about 10% for $K = 2$ (yet still as DT dependent on the offered load) to virtually no burst losses for $K = 20$ only weakly depending on the load (Fig. 5 (c)). Fig. 5 (d) shows the linear decrease of $p_{L,cond}$ with increasing buffer size starting from $K \approx 5$. In Fig. 5 (a) it can be seen that for decreasing buffer size and increasing load, PLoP becomes increasingly unfair (under these conditions the FT share of the number of drops is smaller than the FT bandwidth share of 10%) resulting in relatively less force failures for higher loads (Fig. 6 (a)). For larger values of K ($K \geq 11$), fair operating points are reached. This is due to the fact that the link-speed equivalent buffer is larger^{††} than the voice packet interarrival time of 20ms. Thus a consecutive packet of the *same* flow (which can be surely dropped) can be found with a higher probability in the queue. This allows us to relax assumption 3. of section 2 (see (28)).

To assess whether PLoP can achieve its limited QoS assurance goals with less processing overhead than the other design options given in section 2, we also traced the relative number of queue lookups $\frac{q_l}{a}$ (i.e. searching in the queue and filtering on the flow type, Fig. 6 (b)). This number can be compared against the value $\frac{q_l}{a} = 1$ for design option 2. (2 queues: every packet has to be flow type filtered). It can be seen that except for overload conditions and very low buffer sizes, the overall queue lookup overhead stays roughly below twice the mean FT loss rate (Fig. 5 (a)) for the used drop profile. Additionally, the relative number of full flow ID lookups $\frac{il}{a}$ (=1 for design option 1.: a separate queue for every FT flow) is shown. Again except for overload conditions and very low buffer sizes, the ID lookup overhead (which also indicates the relative number of drop experiments necessary, Fig. 3) stays clearly below 10% (the FT share of the bandwidth).

Fig. 7 (a) shows that background traffic is not negatively affected by PLoP operation in terms of the conditional loss rate. The overall utilization (Fig. 7 (b)), as well as the mean loss rate for BT (not shown here) achieved is equal for either DT and PLoP, because the aggregated loss process (for all flows) has not been changed significantly.

^{††} Assuming a voice packet at the head of the queue and nine H-type BT packets behind it, the time distance (time the voice packet has already been present in the queue under overload) from the head of the queue to the eleventh buffer is $\frac{(9 \times 560 + 208) \times 8 \text{ bit}}{1.92 \times 10^6 \text{ bit/s}} = 21.87 \text{ms}$.

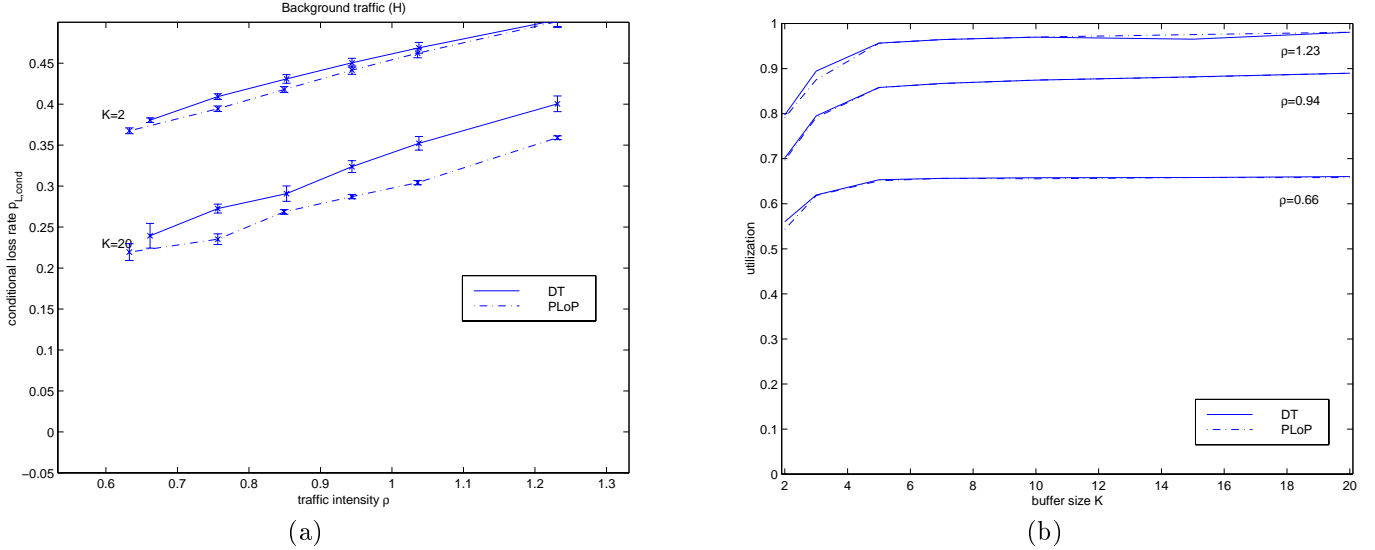


Figure 7. H-type BT conditional loss rate (a), Link utilization (b)

5. CONCLUSIONS

The main result of the paper is that burst loss protection in the data path for periodic traffic (voice) is feasible. The presented PLoP algorithm reduces the conditional loss probability with limited overhead for a wide range of load conditions. If the link-speed equivalent buffer is larger than the expected maximum traffic period, unfairness of the algorithm towards background traffic is avoided. The algorithm operates only during times of congestion and does not require explicit cooperation of the applications. As it enforces a predictive loss pattern, it gives an incentive to build loss-resilient applications (the performance of FEC as well as receiver-only techniques (error concealment) to repair losses is increased). The scheme is useful as an isolated mechanism (especially for potential “last mile” bottlenecks).

Our proposed mechanism aims at influencing the short-term QoS only, without modifying long term throughput. This offers only a weaker incentive to abuse the service. However it may be necessary to monitor misbehaving flows using the proposed basic voice service. Focus of our current research is an implementation of the algorithm into a software-based router to assess the impact of the execution time of the algorithm itself which we did not account for in the presented simulations. Future research includes both an approach of using the “survivor” marking between adjacent PLoP routers as well as mapping the “survivor” marking to Differentiated Services marking. The latter allows a more efficient use of the DS possibilities (marking only packets within “congested areas” of a flow). Future work also includes an integration with DS and IntServ queue management (RED/RIO) and scheduling mechanisms (WFQ, CBQ) possibly serving several flow types (with different flow tables) at the same time. It will be interesting to apply the scheme to other flow types that are sensitive to burst losses but have bursty traffic characteristics. Finally we plan to parametrize existing end-to-end analytic models^{12,37} (which need however to capture adequately the BT burstiness) to reflect the modified end-to-end path characteristics when PLoP is employed.

ACKNOWLEDGMENTS

We are very grateful to the creators of the NS-2 network simulator.³¹ We wish to thank Adam Wolisz, Dorgham Sisalem, Mirko Schramm and Michael Zander for many valuable discussions.

REFERENCES

1. S. Shenker, "Fundamental design issues for the future Internet," *IEEE J. Selected Areas in Communications*, September 1995.
2. D. Reiningger and R. Izmailov, "Soft quality of service with VBR+ video," in *Proceedings of 8th International Workshop on Packet Video (AVSPN97)*, (Aberdeen), September 1997.
3. J. Gruber and L. Strawczyynski, "Subjective effects of variable delay and speech clipping in dynamically managed voice systems," *IEEE/ Transactions on Communications* **Vol. COM-33**, August 1985.
4. J. Suzuki and M. Taka, "Missing packet recovery techniques for low-bit-rate coded speech," *IEEE Journal on Selected Areas in Communications* **Vol. 7**, pp. 707–717, June 1989.
5. V. Hardman, M. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the Internet," in *Proceedings INET '95*, (<http://info.isoc.org/HMP/PAPER/070/abst.html>), 1995.
6. H. Sanneck, A. Stenger, K. B. Younes, and B. Girod, "A new technique for audio packet loss concealment," in *Proceedings IEEE Global Internet 1996 (Jon Crowcroft and Henning Schulzrinne, eds.)*, pp. 48–52, (London, England), November 1996.
7. H. Sanneck, "Concealment of lost speech packets using adaptive packetization," in *Proceedings IEEE Multimedia Systems*, (Austin, TX), June 1998.
8. H. Sanneck, "Adaptive loss concealment for Internet telephony applications," in *Proceedings INET '98*, (Geneva, Switzerland), July 1998.
9. J.-C. Bolot and A. Garcia, "Control mechanisms for packet audio in the Internet," in *Proceedings IEEE Infocom '96*, pp. 232–239, (San Francisco, CA), April 1996.
10. T. Turetletti, S. Fosse-Parisis, and J.-C. Bolot, "Experiments with a layered transmission scheme over the Internet," Technical Report 3296, INRIA, November 1997.
11. C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, Sept./Oct. 1998.
12. N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," in *Proceedings ACM SIGCOMM '90*, pp. 124–131, (San Francisco, CA), June 1990.
13. C. Partridge, "A proposed flow specification," RFC 1363, IETF, September 1992. <ftp://ftp.ietf.org/rfc/rfc1363.txt>.
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*, pp. 355–362, (Tempe/Phoenix, AZ, USA), February 1998.
15. S. Ono, T. Miyata, and H. Fukuda, "Loss metrics of grouped packets for IPPM," Internet Draft, IETF IPPM Working Group, August 1998. <ftp://ftp.ietf.org/internet-drafts/draft-ono-group-loss-00.txt>.
16. R. Koodli and R. Ravikanth, "Impact of loss characteristics on real-time applications," Presentation, Proceedings of the 39th IETF, Washington, DC, USA, December 1997.
17. R. Nagarajan, J. Kurose, and D. Towsley, "Finite-horizon statistical Quality-of-Service measures for high speed networks," *J. High Speed Networks*, December 1994.
18. Z. Liu, P. Nain, and D. Towsley, "Bounds on finite horizon QoS metrics with application to call admission," in *Proceedings IEEE INFOCOM '96*, (San Francisco, CA), April 1996.
19. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," Internet Draft, IETF Diffserv Working Group, August 1998. <ftp://ftp.ietf.org/internet-drafts/draft-ietf-diffserv-arch-01.txt>.
20. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, August 1993.
21. S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," Technical Report, Network Research Group, LBNL, February 1997.
22. R. Guerin, S. Kamat, V. Peris, and R. Rajan, "Scalable QoS provision through buffer management," in *Proceedings ACM SIGCOMM*, (Vancouver, B.C.), September 1998.
23. R. Koodli and C. Krishna, "Supporting multiple-tier QoS in a video bridging application," in *IFIP Fifth International Workshop on Quality of Service (IWQOS '97)*, (New York, NY, USA), May 1997.
24. K. Seal and S. Singh, "Loss profiles: A quality of service measure in mobile computing," *J. Wireless Networks* **Vol. 2**(1), pp. 45–61, 1996.

25. K. Brown and S. Singh, "Loss profiles at the link layer," in *3rd Intl. Workshop on Mobile Multimedia Communication*, September 1996.
26. S. Ramanathan, P. Rangan, and H. Vin, "Frame-induced packet discarding: An efficient strategy for video networking," in *NOSSDAV '93*, pp. 173–184, 1993.
27. J. Rosenberg and H. Schulzrinne, "Issues and options for an aggregation service within RTP," Internet Draft, IETF AVT Working Group, December 1996. <ftp://ftp.ietf.org/internet-drafts/draft-rosenberg-itg-00.txt>.
28. H. Sanneck and G. Carle, "Predictive loss pattern queue management for Internet routers," Technical Report, GMD Fokus, November 1998.
29. Z.Liu and R. Righter, "The impact of cell dropping policies in ATM networks," Technical Report 3047, INRIA, November 1996.
30. H. Schulzrinne, J. Kurose, and D. Towsley, "Loss correlation for queues with bursty input streams," in *Proceedings ICC '92*, pp. 219–224, (Chicago, IL), 1992.
31. *Network simulator ns-2*, October 1998. <http://www-mash.cs.berkeley.edu/ns/ns.html>.
32. 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.
33. P. Newman, T. Lyon, and G. Minshall, "Flow labelled IP: Connectionless ATM under IP," in *Networld + Interop*, (Las Vegas), April 1996.
34. M. Ilvesmäki, K. Kilkki, and M. Luoma, "Packets or ports - the decisions of IP switching," in *Broadband Networking Technologies, Seyhan Civanlar, Indra Widjaja, Editors, Proceedings SPIE Vol.3233*, pp. 53–64, (Dallas, TX), November 1997.
35. M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking* **Vol. 5**, pp. 835–846, December 1997.
36. 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.
37. J.-C. Bolot, H. Crépin, and A. Garcia, "Analysis of audio packet loss in the Internet," in *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 163–174, (Durham, NH), April 1995.