Predictive Loss Pattern Queue Management for Internet Routers

Henning Sanneck and G. Carle
GMD Fokus, Berlin/Germany
{*sanneck,carle*}*@fokus.gmd.de*

SPIE Voice, Video and Data Communications '98
Quality of Service Issues Related to Internet
November 3, 1998, Boston, MA

Introduction

- Motivation: simple, but adequate QoS for Internet Voice
- (Short-term) QoS measures / Drop Profiles

Predictive Loss Pattern ($PLoP$)

- Queue Management Algorithm
- Properties

Evaluation

- Deployment / Test Scenario (Traffic Model)
- Results (performance, necessary overhead)

Conclusions / Future Work

Motivation

- real-time applications (voice !): tolerant to occasional packet loss, sensitive to burst losses

- Integrated Services: *per-flow* QoS setup / state maintenance not scalable (voice), loss tolerance of applications not exploited

- Differentiated Services: *per-packet* QoS; reliance on sender to mark packets, again: worst case assumption made (sender is unaware of amount/location of congestion)

⇒ temporary, local protection of particular flows that have previously been discriminated (i.e. lost packets)
  - no explicit per-flow QoS setup
    (every voice flow needs same minimum QoS, per-flow *type* characterization)
  - simple QoS is applicable to voice
    (simple flow structure: no message (frame) level)

(Short-term) QoS measures

loss burstiness as key QoS parameter $\Rightarrow$ *short-term* QoS measures

- packet arrivals flow i: a_i
- occurence $o_{k,i}$ of a loss run length $k \Rightarrow$ drops: $d_i = \sum_{k=1}^{\infty} k o_{k,i}$
- occurence {"two consecutive packets lost"}: $b_i = \sum_{k=1}^{\infty} (k-1) o_{k,i}$

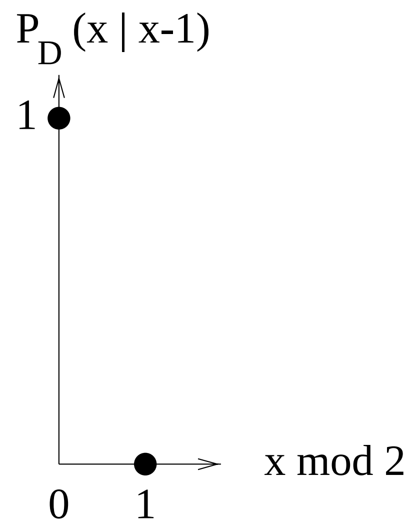
Gilbert model	a_i arrivals	$a_i \rightarrow \infty$
	$p_{L,i} = \frac{d_i}{a_i}$ mean loss rate	$P(X = loss)$ unconditional loss prob.
conditional loss	$p_{L,cond,i} = \frac{b_i}{d_i}$	$P(X = loss \mid X = loss)$

Limitations

- no information about time-relation of loss bursts ("burstiness of bursts")
- longer bursts ($k > 2$) mapped into b_i

Drop Profiles

Translation of the applications' end-to-end QoS requirements (i.e. minimization of $p_{L,cond,i}$) to a *per-packet* behaviour of a queue management algorithm at a single node



Drop profile for voice

$\Rightarrow$ $p_{L,cond,i} = 0$, bound on mean loss rate $\hat{p}_{L,i} = 0.5$
$\Rightarrow$ Design of a queue management algorithm operating at $p_{L,i} \ll \hat{p}_{L,i}$, $p_{L,cond,i} \to 0$

Predictive Loss Pattern ($PLoP$)

Enforcement of the drop profile for a group of flows (foreground traffic [FT]: Voice)

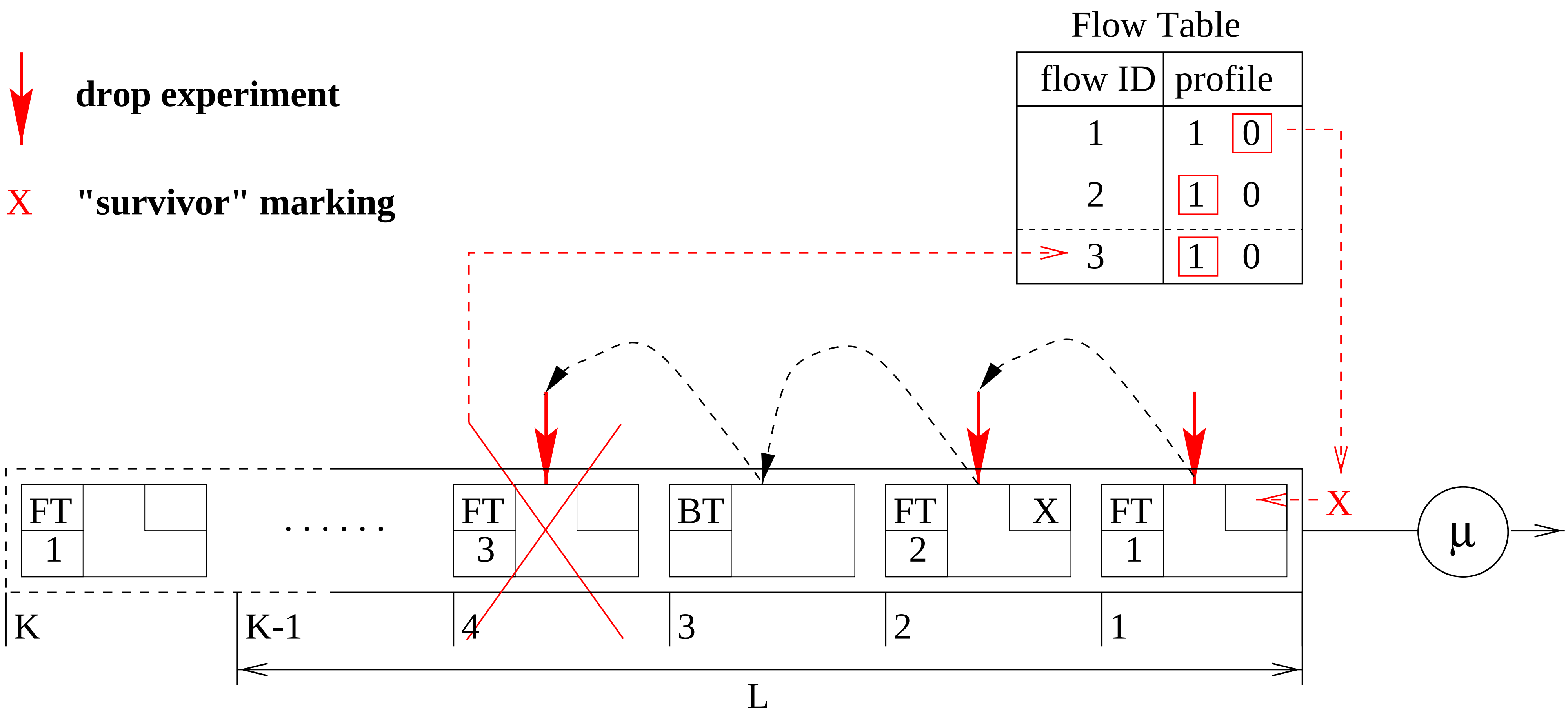
- queue occupancy of single FT flow might be < 2 packets
- FT non-bursty (*mean* occupancy of aggregated FT flows changes slowly)

$\Rightarrow$ shift packet drop which would violate the profile for flow i to flow $j \in FT, j \neq i$

- need to keep *partial* per-flow state only
- queue management (instead of classifier/scheduler)

$\Rightarrow$ simplicity, scalability

PLoP operation



$\frac{L}{\mu} > T_i$: higher probability that replacement packet (of same flow) can be found
($\frac{L}{\mu}$: link-speed equivalent buffer, T_i: interarrival period flow i)

Queue Management Algorithm

PLoP()

```
if queue threshold exceeded
    if (packet ∈ FT)   [flow type filter]
        status = drop_experiment()
        if (status == FAILED)
          [force failure]
          drop
    else
        drop
```

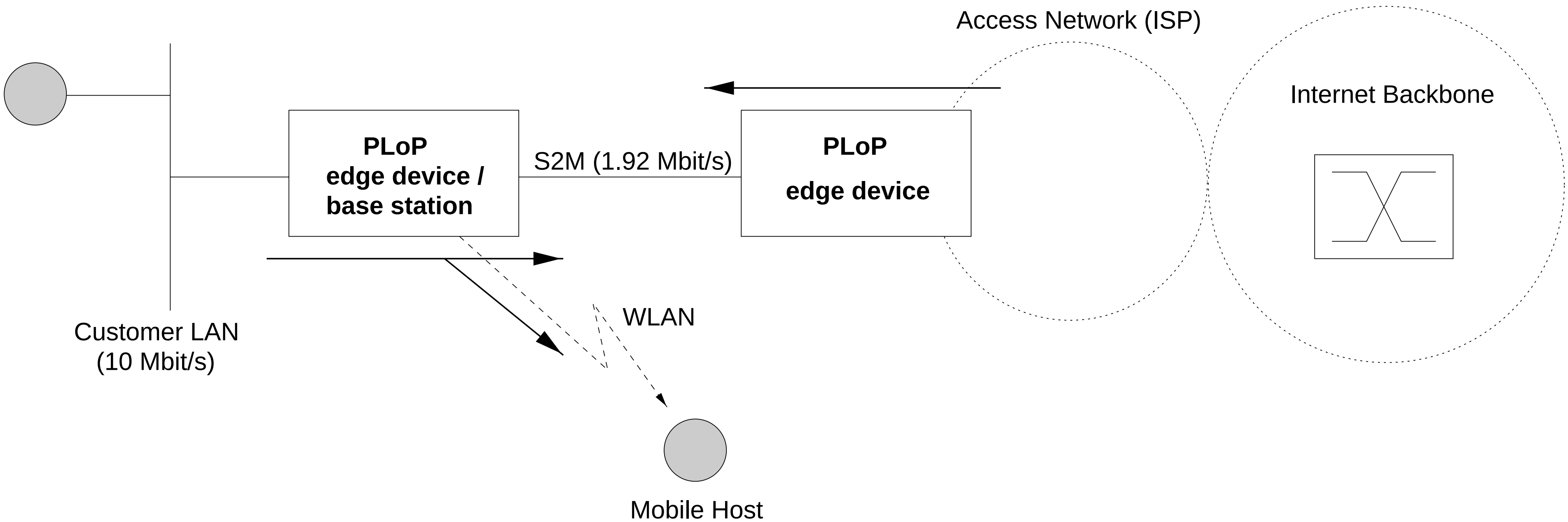
drop_experiment()

```
if (flow not in flow table)   [flow ID filter]
    create flow table entry
generate random number R ∈]0 , 1]
if R ≤ 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
```

PLoP Properties

- **choice of dropping discipline / search direction**
  drop front / *search front* (otherwise accumulation of “survivors”)
- limit on maximum flow table size ($\hat{p}_{L,i}$!)
- **flow table management policy**
  - *preemptive* (FIFO), one timer (congested/uncongested)
  - non-preemptive, per-entry timer (other FT flows “blocked”)
- **force failure policies**
  - *drop front FT packet*
  - drop other BT (which packet ?)

Deployment / test scenario

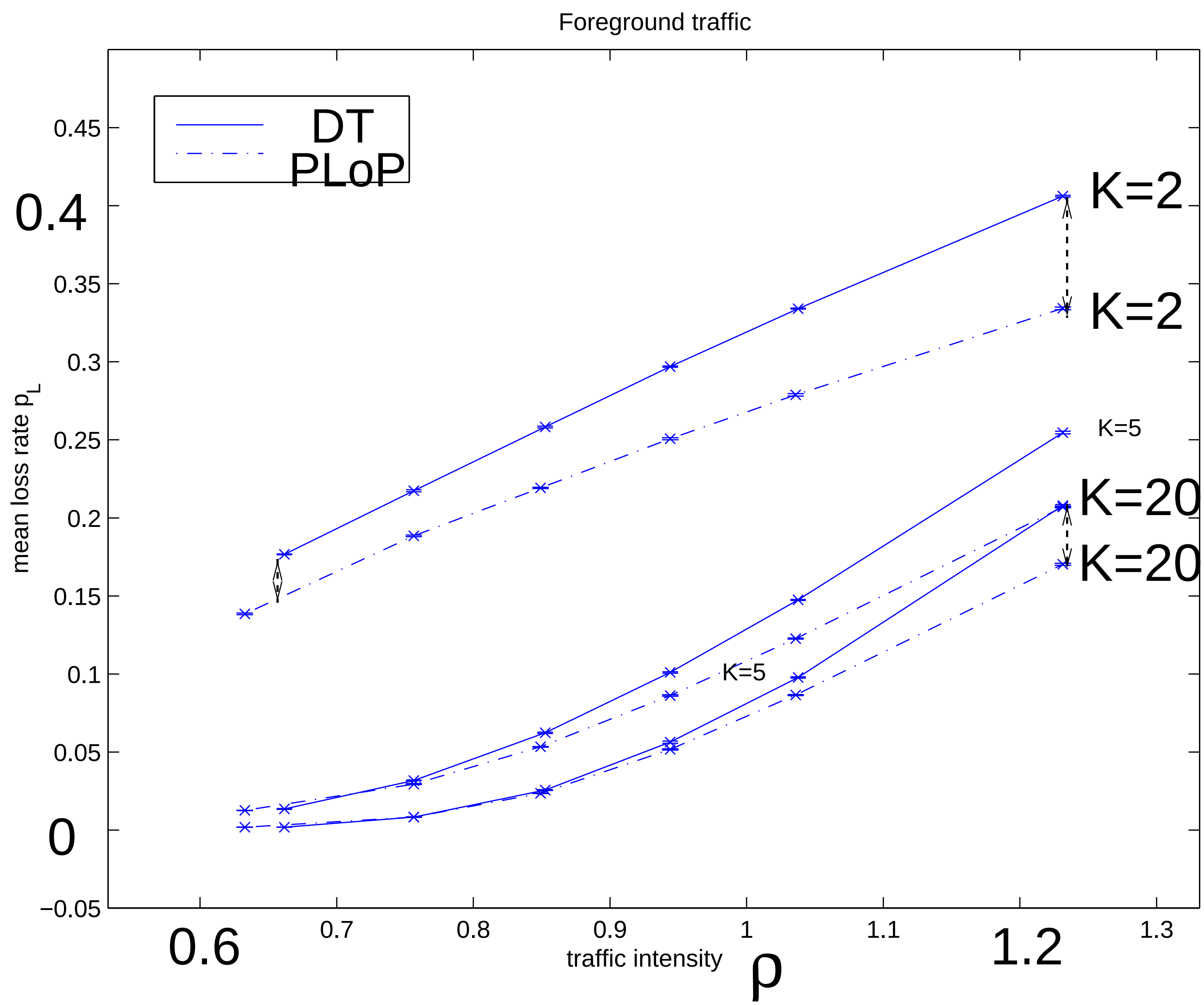


Traffic Model

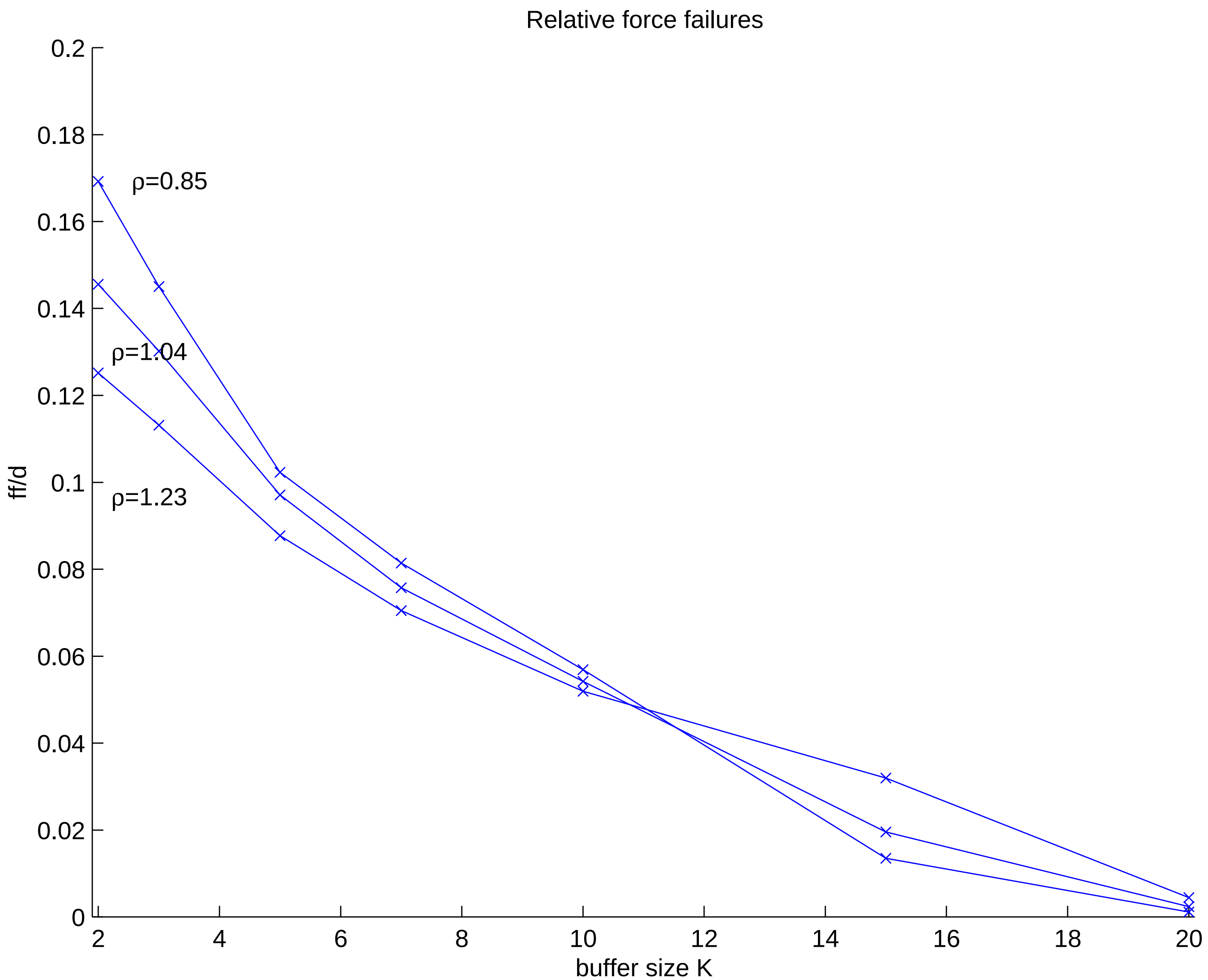
	"http" BT	*"dns" BT*	*FT (voice)*
bandwidth share ($\rho = 1$) (% of gateway b/w)	80	10	10
flow share (% of BT)	75	25	-
active flows ($\rho = 1$)	18	6	6
peak bandwidth ($\frac{kBit}{s}$)	256	30...34	83.2
packet size (bytes)	48+512	36+92	48+160
on/off distribution	Pareto (shape=1.9)	Exponential	Exponential
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

Experiment: variation of BT (LRD traffic) intensity for various buffer sizes
simulation time ≈ 14 h for each run, ns-2 network simulator

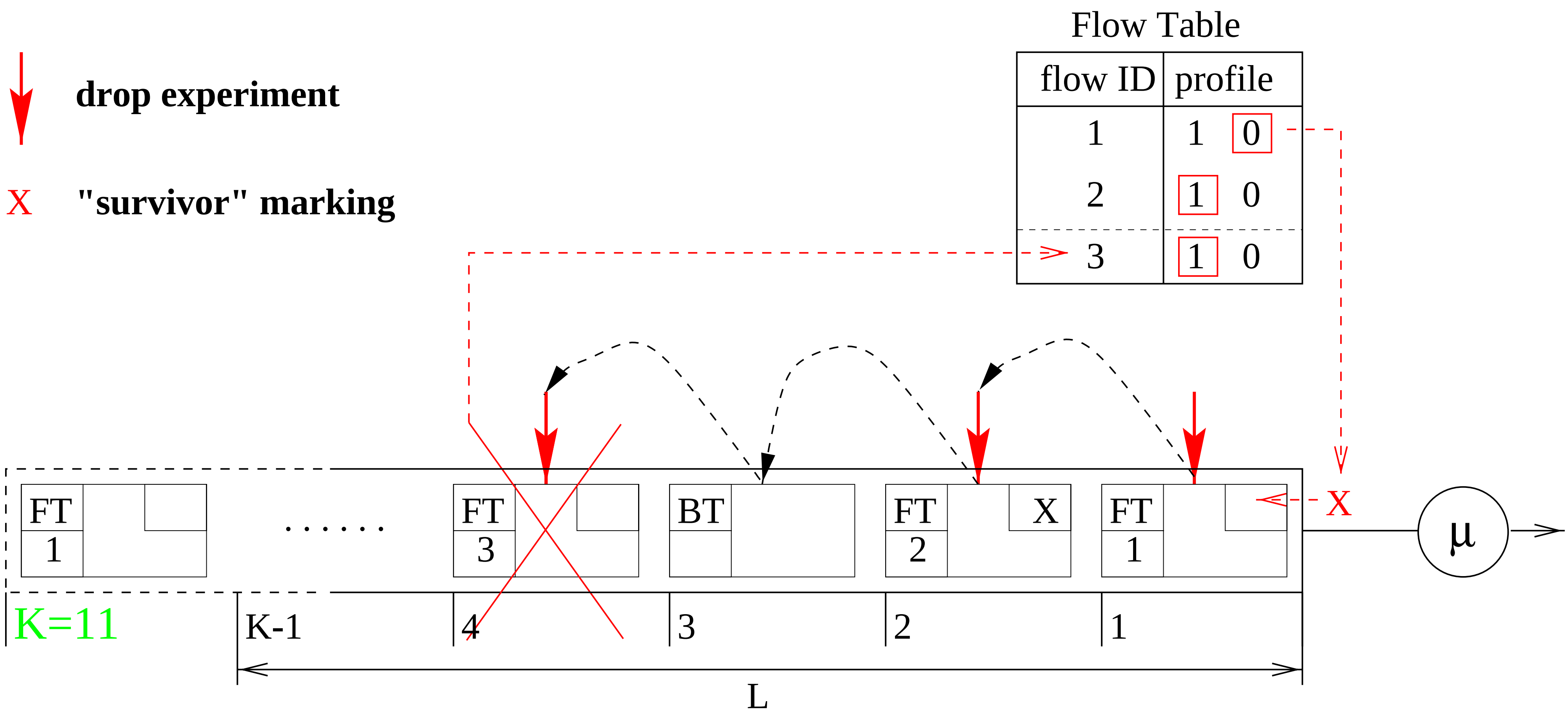
Mean Loss Rate p_L vs. traffic intensity $\rho = \lambda/\mu$



Force Failures ff/D vs. buffer size K

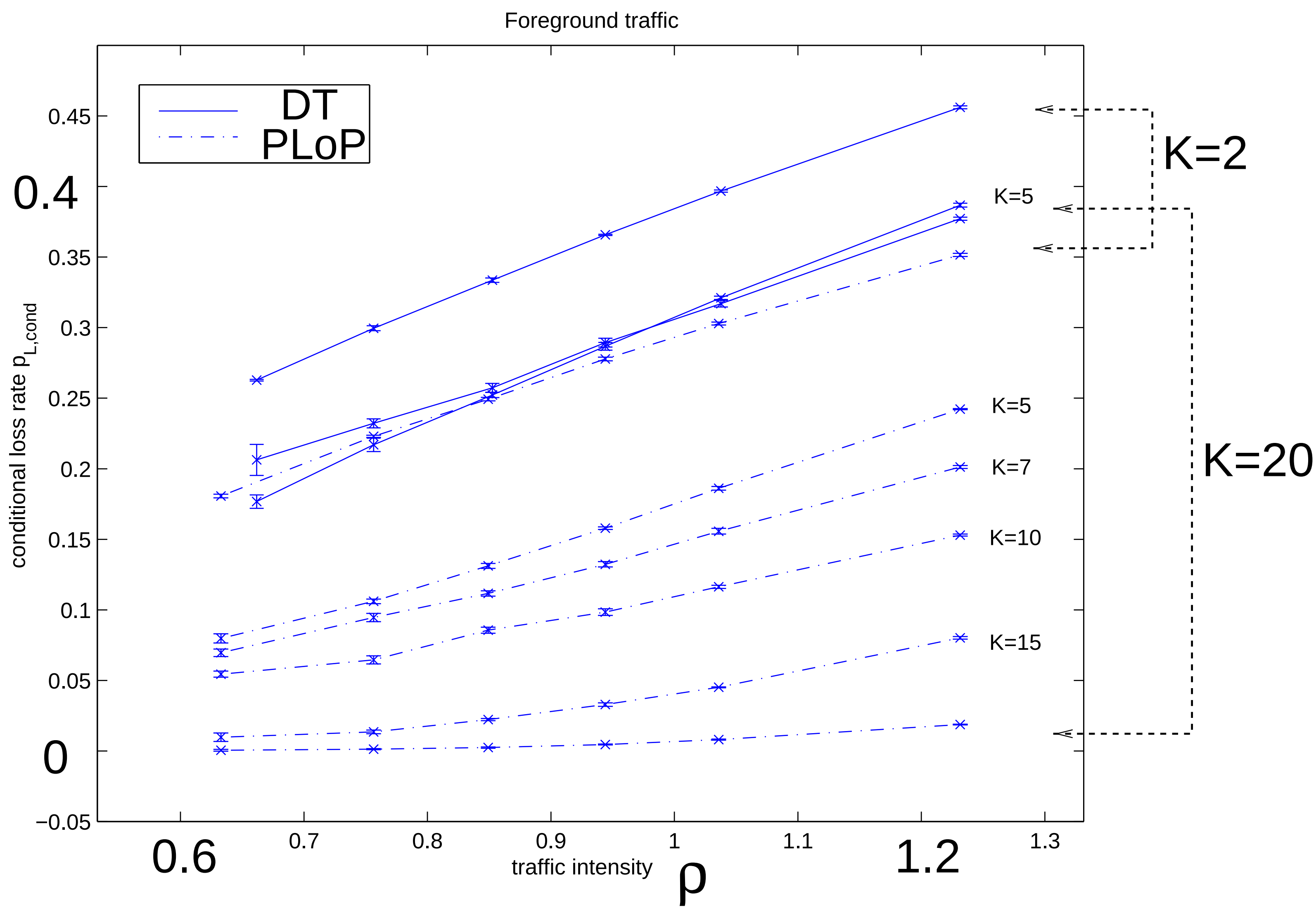


PLoP operation

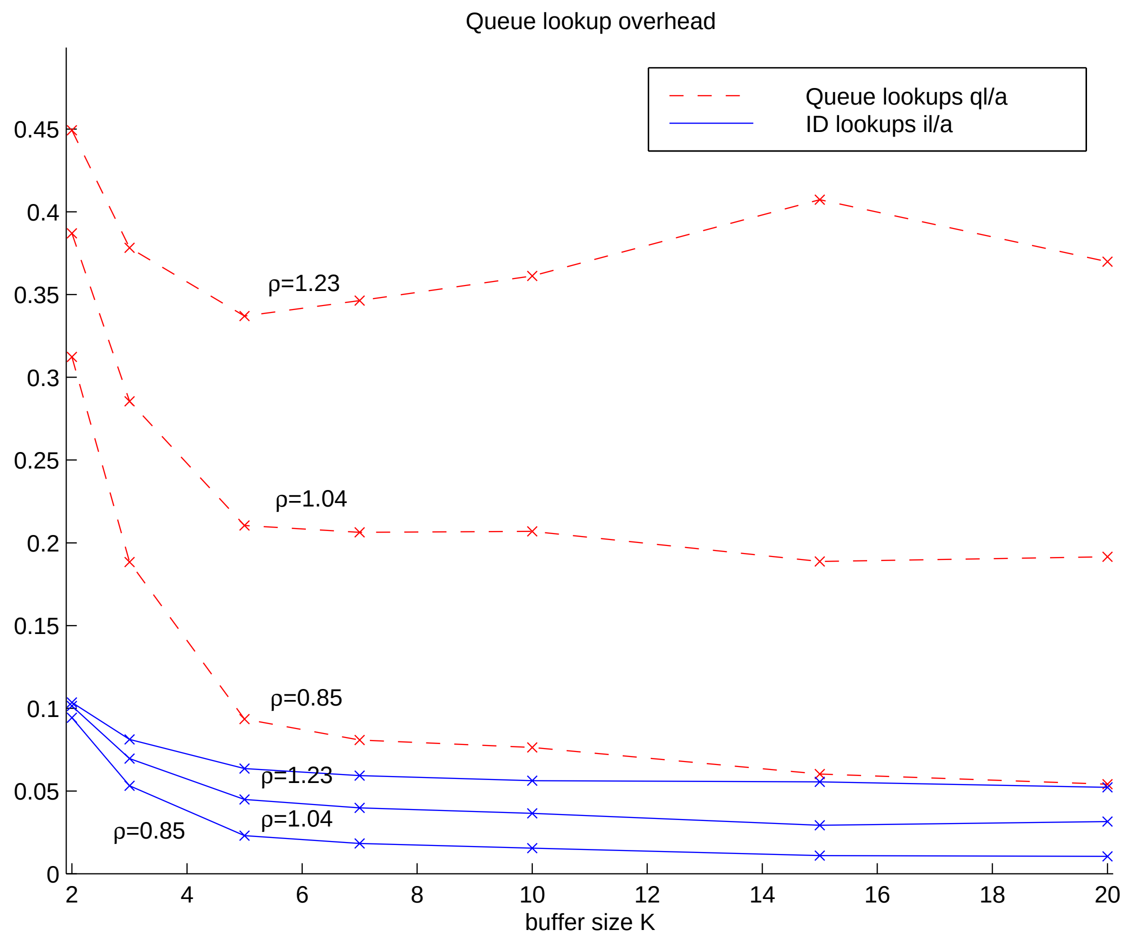


$\frac{L}{\mu} > T_i$: higher probability that replacement packet (of same flow) can be found
($\frac{L}{\mu}$: link-speed equivalent buffer, T_i: interarrival period flow i)

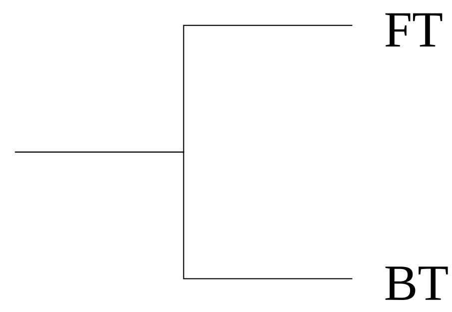
Conditional Loss $p_{L,cond}$ vs. traffic intensity ρ



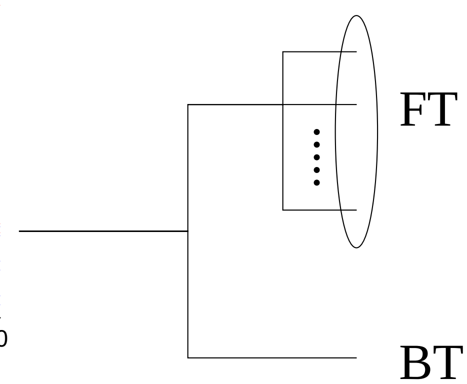
FT/BT lookups ql/A, ID lookups il/A vs. buffer size K



ql/A = const. = 1

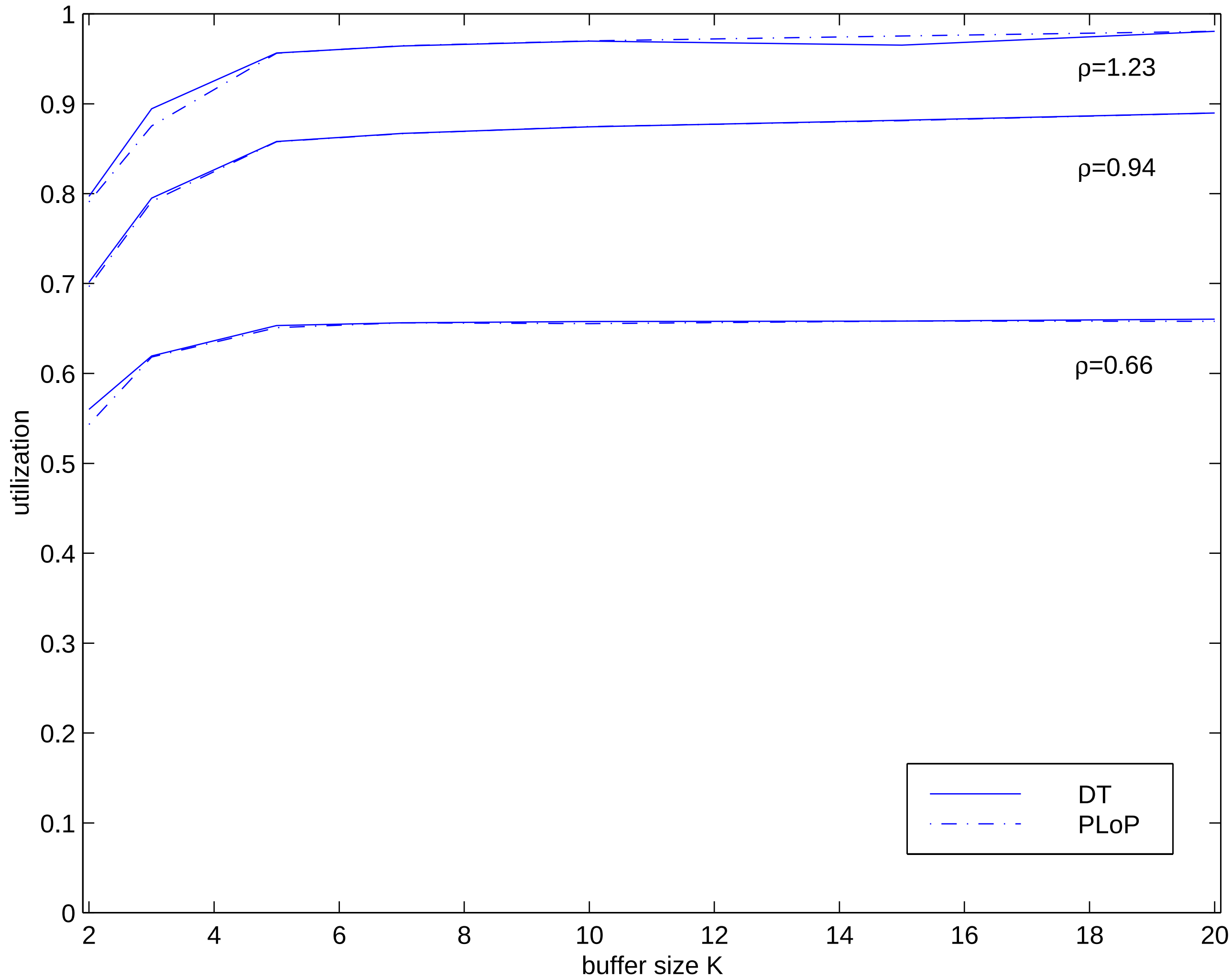


il/A = f(FT share) = 0.1

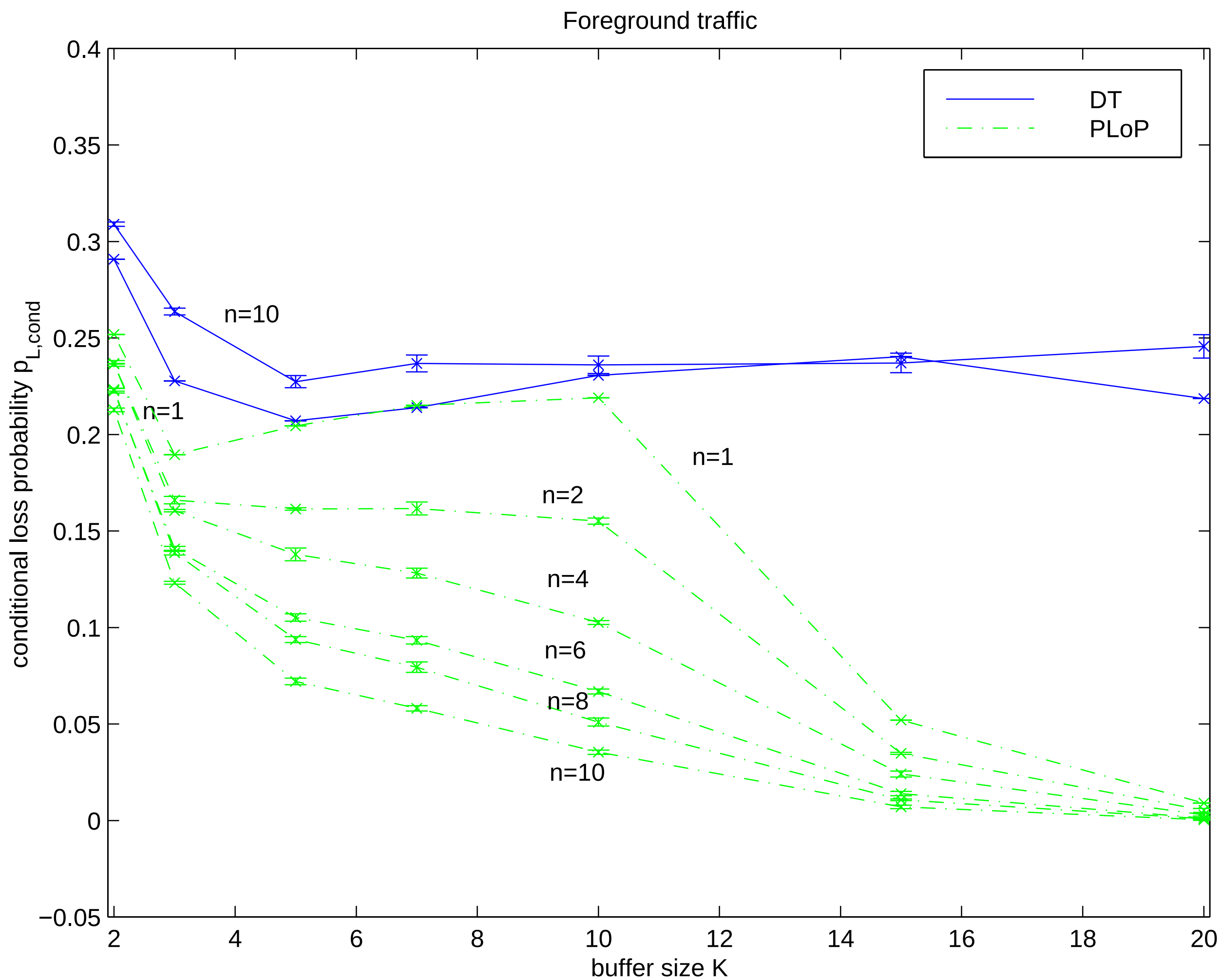


Relative number of lookups = f(FT loss)

Link Utilization vs. buffer size K



Conditional Loss $p_{L,cond}$ vs. buffer size K



n: number of active FT flows

Conclusions / Future work

- burst loss protection in the data path for periodic traffic (voice) is feasible (reduction of $p_{L,cond,i}$ with limited overhead for a range of load conditions)
- link-speed equivalent buffer $>$ maximum traffic period
  $\Rightarrow$ unfairness of the algorithm towards BT is avoided.
- operates only during times of congestion
  useful as isolated mechanism ("last mile" bottleneck)
- does not require explicit cooperation of the applications
  enhances application-level end-to-end loss recovery
- implementation ongoing $\rightarrow$ assess the impact of the algorithm execution time
- distribution of drop profiles needs to be addressed

Window-based loss prob. b/a vs. traffic intensity ρ