Byte and Packet Congestion Notification

draft-briscoe-tsvwg-byte-pkt-mark-00.txt

Bob Briscoe, BT & UCL

IETF-69 tsvwg Jul 2007

initial draft

- Byte and Packet Congestion Notification
  - **initial draft:** draft-briscoe-tsvwg-byte-pkt-mark-00.txt
  - **intended status:** informational
  - **immediate intent:** move to WG item

exec summary

- adjust for bytes when transport reads NOT when network writes
  - i.e. byte-size of packets notifying congestion (whether dropped or ECN marked)
- byte-mode packet drop (small pkt → lower drop prob)?
  - AQM / RED RFC2309 (sort-of) recommends it
  - propose ‘SHOULD NOT’ to avoid perverse incentive to create small packets
  - survey of >80 vendors (~20% responded): none implemented anyway

---

- NOTE: only ‘byte mode packet drop’ deprecated
  - ‘byte mode queue measurement’ (often called just ‘byte mode’) is OK

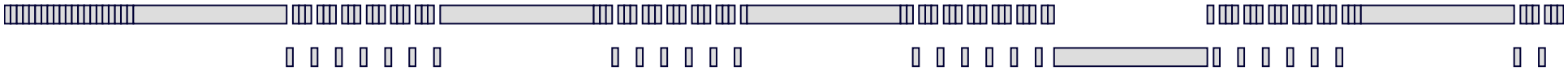
example: comparing each RED mode

simple packet streams (no congestion response)

RED packet-mode packet drop

- same drop probability for any packet
- universally deployed
- propose: SHOULD

	1500B pkts	60B pkts
input	1Mbps	1Mbps
drop prob.	25%	25%
output	750kbps	750kbps



RED byte-mode packet drop

- lower drop probability for smaller packets
- 'RED' RFC2309 (sort of) recommends
- propose: SHOULD NOT

	1500B pkts	60B pkts
input	1Mbps	1Mbps
drop prob.	25%	1%
output	750kbps	990kbps

layer to adjust rate for size of a dropped packet

network or transport?

✘ network layer adjustment

transport congestion control	packet-mode packet drop	linear byte-mode packet drop	squared byte-mode packet drop
TCP [RFC2581] or TFRC [RFC3448]	$\frac{s}{\sqrt{p}}$	$\frac{s}{\sqrt{p.s}} = \frac{\sqrt{s}}{\sqrt{p}}$	$\frac{s}{\sqrt{p.s^2}} = \frac{1}{\sqrt{p}}$
TFRC-SP [RFC4828]	$\frac{1}{\sqrt{p}}$	$\frac{1}{\sqrt{p.s}} = \frac{1}{\sqrt{s}\sqrt{p}}$	$\frac{1}{\sqrt{p.s^2}} = \frac{1}{s\sqrt{p}}$

✓ transport layer adjustment

flow bit rate per RTT in terms of
s = packet size
p = drop (or marking) rate prior to adjustment

RED byte mode packet drop deployment survey

14	17%	not implemented
2	2%	not implemented probably (tbc)
0	0%	implemented
68	81%	no response (so far)
84	100%	companies/org's surveyed

- wide range of types of company
  - large L3 & L2 equipment vendors
  - wireless equipment vendors
  - firewall vendors
  - large software businesses with a small selection of networking products
- "no response" includes 10 open source (Linux/FreeBSD) institutions
  - quick look at one (Fedora): not implemented
- "not implemented" includes very large fraction of the market
  - e.g. Cisco, Alcatel-Lucent (two who have given permission to be identified)
- since 10-Nov-2004 implemented as default in ns2 simulator
  - **NOTE:** later ns2 simulations with mixed packet sizes may not be representative of real Internet

why change advice now?

- DCCP
  - e.g. TFRC small packet variant experiment [RFC4828]
- PCN marking algorithm design
  - imminent (chartered)
- RED implementations; deployed & new
  - prevent giving perverse incentives to create small packets

tsvwg WG item?

- no time for...
  - distinguishing byte-congestible & packet congestible (open research issue) – **see I-D**

NOTE: don't turn off RED completely: favours small packets

- at least as much as RED byte mode packet drop

NOTE again: only byte mode packet drop deprecated

- byte mode queue measurement (often called just 'byte mode') is OK

Byte and Packet Congestion Notification

draft-briscoe-tsvwg-byte-pkt-mark-00.txt

Q&A

bit-congestible and packet-congestible

- bit-congestible resources
  - e.g. transmission links, buffer memory
- packet-congestible resources
  - e.g. route look-up, firewall
- focus on bit-congestible only
  - by design packet processors typically protected by bit-rate limits
- stages where byte-size might be relevant:
  1. measuring congestion (queue length in bytes or packets?)
  2. coding congestion (drop or ECN marking) into a packet
  3. decoding congestion from a packet
- #1 is well understood and orthogonal
- we'll focus on #2 vs. #3