Packet Size & Congestion Control

draft-briscoe-tsvwg-byte-pkt-mark-02.txt

Bob Briscoe, BT & UCL

IRTF ICCRG Mar 2008

what does congestion notification on a packet of a certain size mean?

- notification of excess bits?
  - transport reduces bit-rate
- notification of excess packets?
  - transport can increase packet size but hold bit-rate
- neither of the above?

for any of:

- drop
- ECN
- PCN [PCN]
- deterministic marking [DPM, ADPM]
- Δexplicit rates (e.g. XCP)

related questions

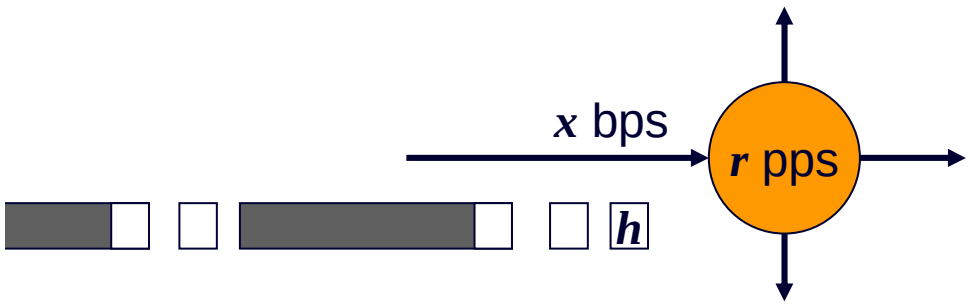
- how should congestion notification scale with packet size?
  - principles for future protocol design taking into account existing deployments
- which algorithms should depend on packet size?
  - when network equipment encodes congestion notification into a packet?
  - and/or when transport decodes congestion notification from a packet?

why decide now?
between transport & network

- part of answering ICCRG question
  - what's necessary & sufficient forwarding hardware for future cc?
- near-impossible to design transports to meet guidelines [RFC5033]
  - if we can't agree whether transport or network should handle packet size
- DCCP CCID standardisation
  - hard to assess TFRC small packet variant experiment [RFC4828]
- PCN marking algorithm standardisation
  - imminent (chartered) but depends on this decision
- what little advice there is in the RFC series (on RED) is unclear:
  - it seems to give perverse incentives to create small packets
  - it seems to encourage a dangerous DoS vulnerability
- evolving larger PMTUs may solve other scaling problems

bit-congestible and packet-congestible

- bit-congestible resources
  - e.g. transmission links, most buffer memory
- packet-congestible resources (often cycle-congestible)
  - e.g. route look-ups, firewalls, fixed size packet buffers
- most network resources are solely bit-congestible
  - by design, max bit-rates protect packet processors
  - (no survey evidence for this – only assertions)

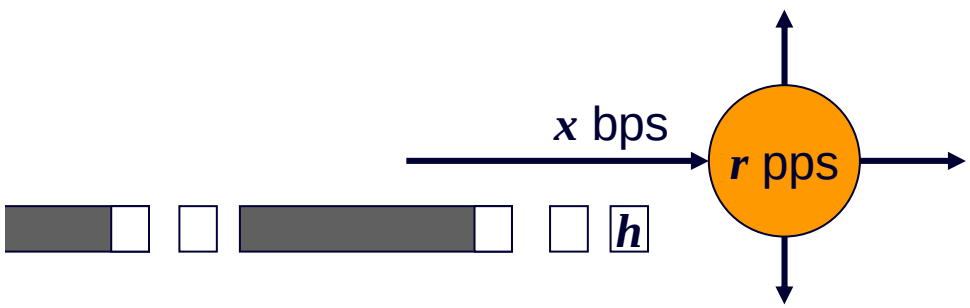


consider a link of bit-rate x [bps]
feeding a packet processor of rate r [pps]
with min packet size of h [b/pkt]

as long as $r \geq x/h$
resource is always bit-congestible

increasing *range* of packet sizes

- as we increase max packet size to increase bit-rate
  - *min* packet size doesn't increase too
  - cannot guarantee transports will *not* send tiny packets
- future could be more mixed
  - bit-congestible & packet-congestible
  - but processing speed growth currently faster than transmission



as x increases with h const
if growth in r doesn't keep up
$r \geq x/h$ may no longer hold
resource sometimes pkt-congestible?

growing list of confusable causes of drop

1. transmission loss
2. congestion
   a) bit-congestion
   b) packet-congestion
3. policing
   a) for numerous reasons
   b) …beyond scope today

- if we find a way to distinguish 1. & 2.,
  when standardising we should consider distinguishing 1, 2a), 2b), 3)...

- safe approach
  - if unsure, assume byte-congestion and reduce bit-rate (& pkt-rate)
  - only maintain bit-rate if explicit indication otherwise wholly explains losses

future protocol design

cause of a drop will remain unguessable

- not cost-effective for all resources to include smarts
  - AQM, XCP, etc will never be omnipresent
  - consider higher layer devices: firewalls, servers, proxies and lower layer devices: home-hubs, DSLAMs, WLAN cards, node-Bs
- careful network design can hide dumb queues
  - so even worst traffic matrix cannot congest dumb queues (spare slide)
    - sufficient overprovisioning of dumb resources
    - upstream elements contain AQM smarts: ‘sacrificial throttling’
- but transports cannot assume careful network design
  - AQM has to remain an optimisation, not a generic invariant

which layer should adjust for packet size

network or transport?

- stages where packet size might be relevant:
  1. measuring congestion (queue length in bytes or packets?)
  2. coding congestion (drop or ECN marking) into a specific packet
  3. decoding congestion notification from a specific packet
- #1 is orthogonal to others
  - only depends on how the resource gets congested
  - complicated (see I-D [byte-pkt]) but not controversial
  - local implementation issue, not IETF/IRTF standards
- we'll focus on #2 vs. #3

tempting to reduce drop for small packets

- drops less control packets, which tend to be small
  - SYNs, ACKs, DNS, SIP, HTTP GET etc
- makes TCP bit-rate less dependent on pkt size

- but we need principles – these are merely expedients

- small != control
  - favouring smallness will encourage smallness
- given TCP's bit-rate depends on packet size
  - is that sufficient reason to change the network layer for every transport?

proposed test

congestion control scaling with packet size

- two scenarios: identical except for one aspect
- same number of sources with same mix of apps divide the same load into
  1. fewer large packets
  2. more small packets
- passes if it responds to congestion in the same way in both scenarios

---

- assume links shared by many flows
  - increasing congestion hits more flows with drops/marks

does reducing drop for small packets scale?

- byte-mode drop variant of RED
  - for bit-congestible resources FAILS scalability test
    - even combination of TCP & squared byte-mode RED [Cnodder] which cancels out dependence on packet size of TCP's bit rate

- intuition
  - as packet sizes increase, the higher drop fraction needed to get the same bit-rate removes an increasing fraction of the goodput, requiring greater load to compensate
  - conversely, with smaller packets, very few bytes need to be dropped to notify TCP with sufficient packets. So when queues actually overflow, the bytes that have to be discarded represent a much higher notification fraction, causing TCP to overreact

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]	$\frac{s}{\sqrt{p}}$	$\frac{s}{\sqrt{p.s}} = \frac{\sqrt{s}}{\sqrt{p}}$	$\frac{s}{\sqrt{p.s^2}} = \frac{1}{\sqrt{p}}$
✓ transport layer adjustment: TFRC-SP [RFC4828]	$\frac{1}{\sqrt{p}}$	$\frac{1}{\sqrt{p.s}} = \frac{1}{\sqrt{s}\sqrt{p}}$	$\frac{1}{\sqrt{p.s^2}} = \frac{1}{s\sqrt{p}}$

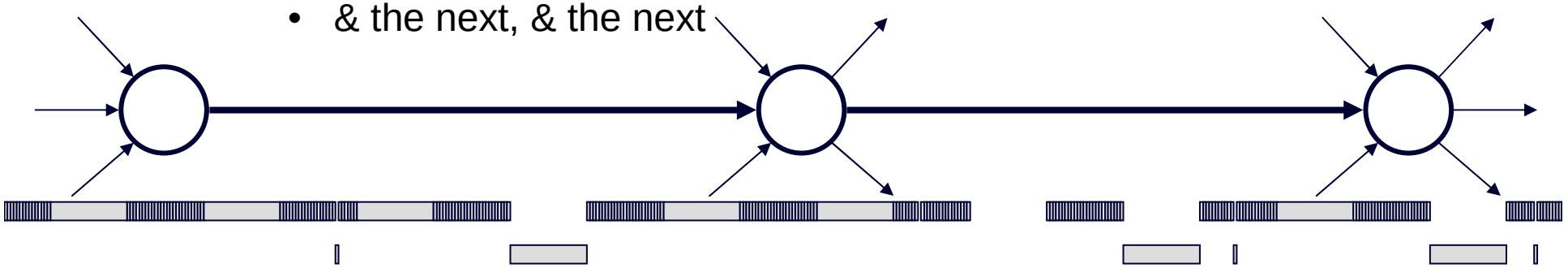
flow bit rate per RTT in terms of

s = packet size

p = drop (or marking) rate prior to adjustment

favouring small packets:
DoS vulnerability

- small packet attacks push out larger packets
  - leaving most small packets to attack the next queue
  - & the next, & the next



- DoS vulnerability similar to that of drop tail queues
- AQM was partly about not locking-out large packets*
  - shouldn't add lock-out back again in the AQM algorithm

* not stated and not a motivation according to at least one author (Floyd)

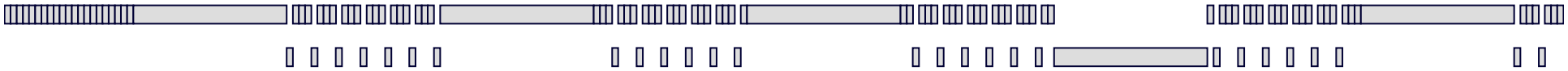
example: comparing each RED mode

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.	48%	1.9%
output	520kbps	980kbps

see note in I-D about dynamic effects

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 byte-mode RED default in ns2 simulator
  - **NOTE:** later ns2 simulations with default RED & mixed packet sizes likely to be very unlike real Internet

summary

congestion notification on a packet of a certain size means...

- ...notification of excess bits
  - assuming a predominantly bit-congestible world
- open research question: is a packet-congestible world likely?
  - pls discuss on iccrg@cs.ucl.ac.uk
- need consensus: allow for packet size in transport, not network
  - AQM algorithms should not favour small packets*
  - pls discuss / support / bash this I-D on tsvwg@ietf.org
- need a programme of transport congestion control updates
  - to take this meaning of packet size into account
  - to ensure transports (including TCP) scale with packet size

---

* don't turn off RED completely: would also favour small packets
  - at least as much as RED byte mode packet drop

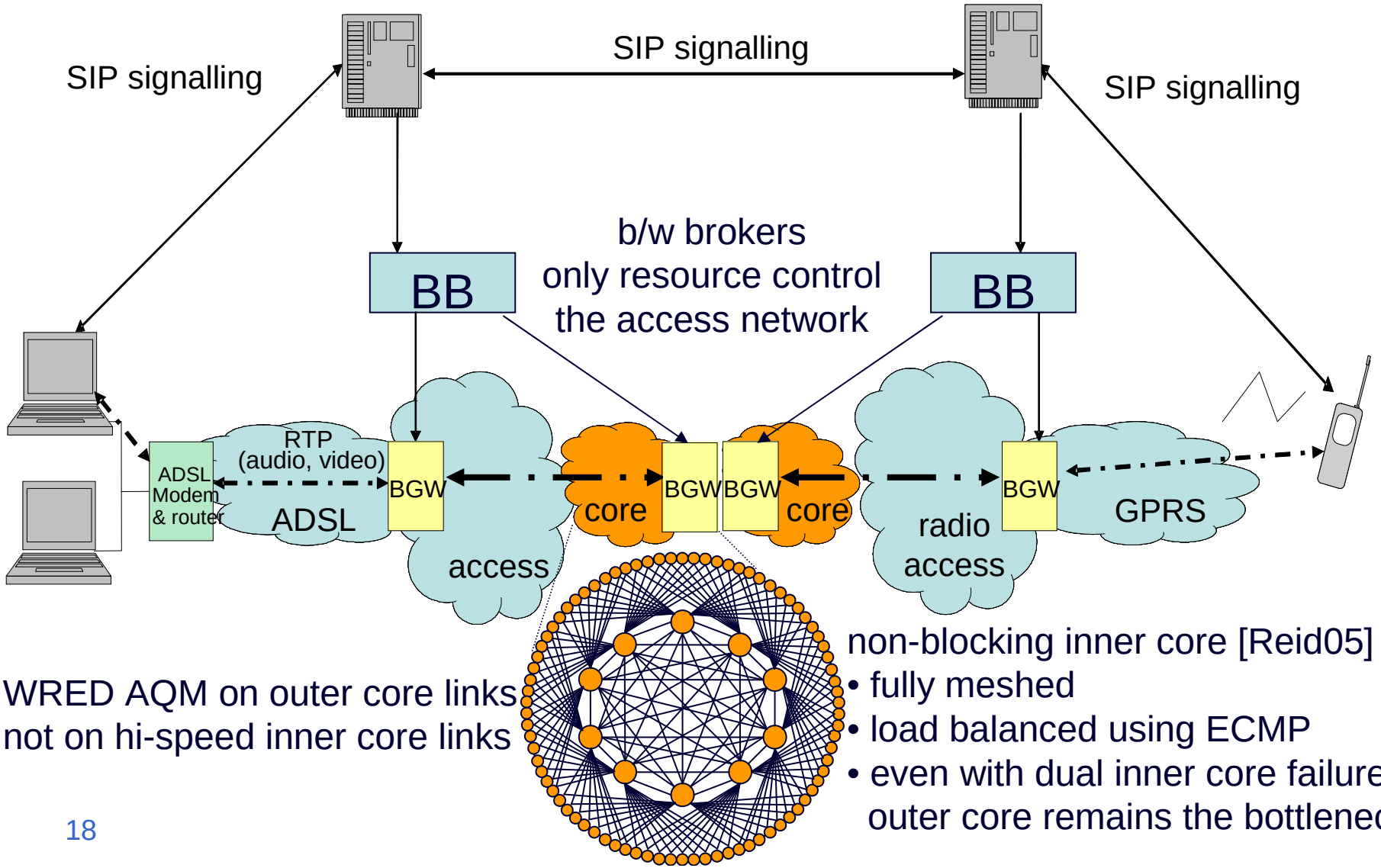
* only RED byte mode packet drop deprecated
  - byte mode queue measurement (often called just 'byte mode') is OK

Packet Size & Congestion Control

draft-briscoe-tsvwg-byte-pkt-mark-02.txt

Q&A

sacrificial throttling: example



- WRED AQM on outer core links
- not on hi-speed inner core links

non-blocking inner core [Reid05]

- fully meshed
- load balanced using ECMP
- even with dual inner core failures
  outer core remains the bottleneck

more info

(not including the well-known stuff)

[byte-pkt] Bob Briscoe, “Byte and Packet Congestion Notification” draft-briscoe-tsvwg-byte-pkt-mark-02.txt (work in progress), (Feb 2008)

[Reid05] Andy B. Reid, Economics and scalability of QoS solutions, BT Technology Journal, 23(2) pp97 – 117 (April 2005)

[PCN] Eardley, P., “Pre-Congestion Notification Architecture,” draft-ietf-pcn-architecture-03 (work in progress), (Feb 2008)

[RFC2039] Bob Braden et al “Recommendations on Queue Management and Congestion Avoidance in the Internet,” RFC 2309 (Apr 1998)

[RFC4828] Floyd, S. and E. Kohler, “TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant,” RFC 4828 (Apr 2007)

[ADPM] Lachlan Andrew et al, “Adaptive Deterministic Packet Marking” *IEEE Comm. Letters*, 10(11):790-792 (Nov 2006)

[DPM] R.W. Thommes and M.J. Coates, “Deterministic Packet Marking for Time-Varying Congestion Price Estimation”, IEEE/ACM Transactions on Networking, 14(3):592-602 (Jun 2006)