# solving this Internet resource sharing problem... and the next, and the next

IETF
P2P infrastructure workshop

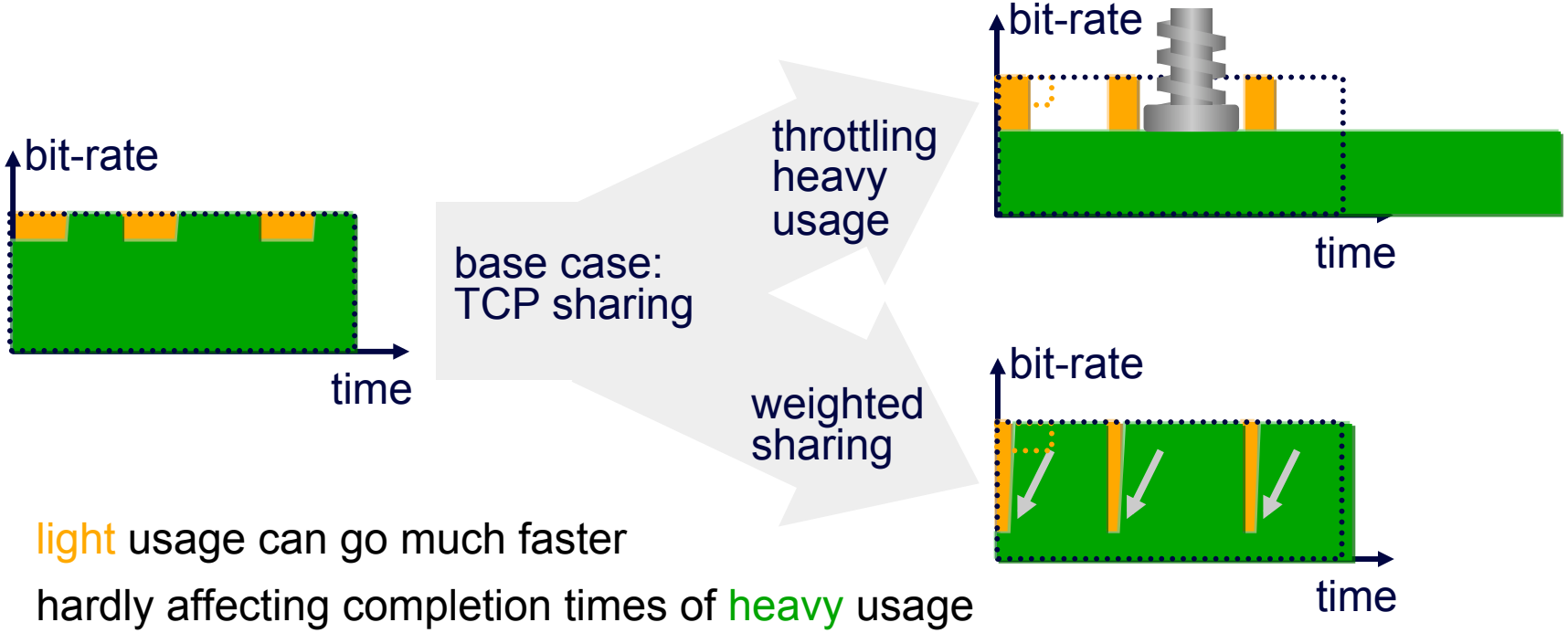
**Bob Briscoe**
**Chief Researcher**
**BT Group (& UCL)**
Lou Burness, Toby Moncaster, Phil Eardley, BT

part-funded by Trilogy, supported by the EC

May 2008



# there are better solutions than fighting

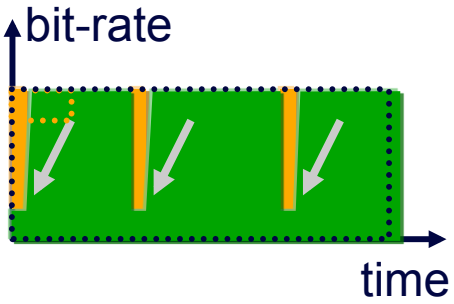



- light usage can go much faster
- hardly affecting completion times of heavy usage

## a challenge

- can the IETF put all the pieces in place to make this happen?
- and make it as simple to deploy as a DPI box?

BT




## weighted sharing

# two means to the same end

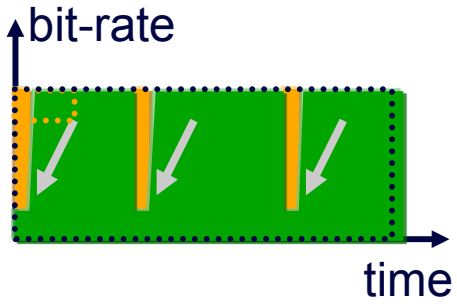

- diffserv weighted scheduling
  - background class at lower scheduling weight than best effort
- weighted congestion control
  - gets ***w*** times more share of bottleneck than a flow with ***w=1***
  - ***w<<1*** for background, ***w>>1*** for interactive

IETF shouldn't pre-judge balance of control

BT

## weighted sharing

# deployability of either?




<u>diffserv weighted scheduling</u>

- APIs:
  - detecting usable classes (1st hop)
  - selecting a class
  - consensus?
- control
  - initially DPI will probably decide
  - even with API, is network complying?
- policing
  - DS not designed for individual users
  - by volume? time-of-day? onnet-offnet?
  - won't peak period shift once declared?
  - DS sender-based? *down*load limits?
- interconnect
  - interactions: light user sending to heavy?
- moving the problem
  - heavy hitters within either/both classes?

<u>weighted congestion control</u>

- APIs:
  - -
  - setting the weight
  - consensus?
- control
  - unambiguously host controlled
  - network control all in the policing...
- policing
  - why not always set weight to max?
  - policing by what metric?
  - will high weight bursts harm r-t apps?
  - sender-based? *down*load limits?
- interconnect
  - why police congestion of other ISPs?

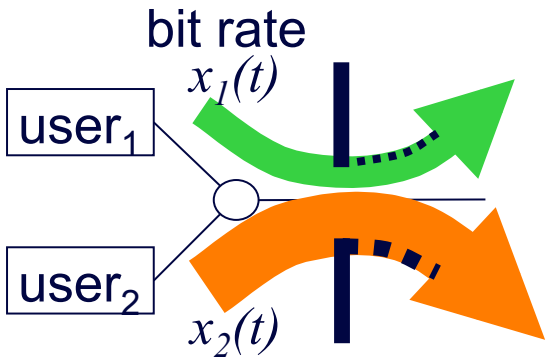
BT

# features of good solutions

- sending is sender's responsibility at the network layer
  - receiver responsibility for asking to be sent to is for higher layers
- user control within an envelope
  - ISP-enforced envelope sufficient to protect other customers
  - within envelope: full user control of priorities proves ISP neutrality
  - ISP can *offer* prioritisation by DPI – optional service, not imposition
- no need for wriggle-room
  - ISP currently final judge of acceptable use policy – has to be woolly
  - because using volume doesn't really measure harm to others
  - leaves room for later interpretation – breeds suspicion & conflict
- each ISP free to choose not to do this
  - but an ISP who wants to, can protect itself from the users of ISPs who don't

## core of solution

# congestion harm (cost) metric



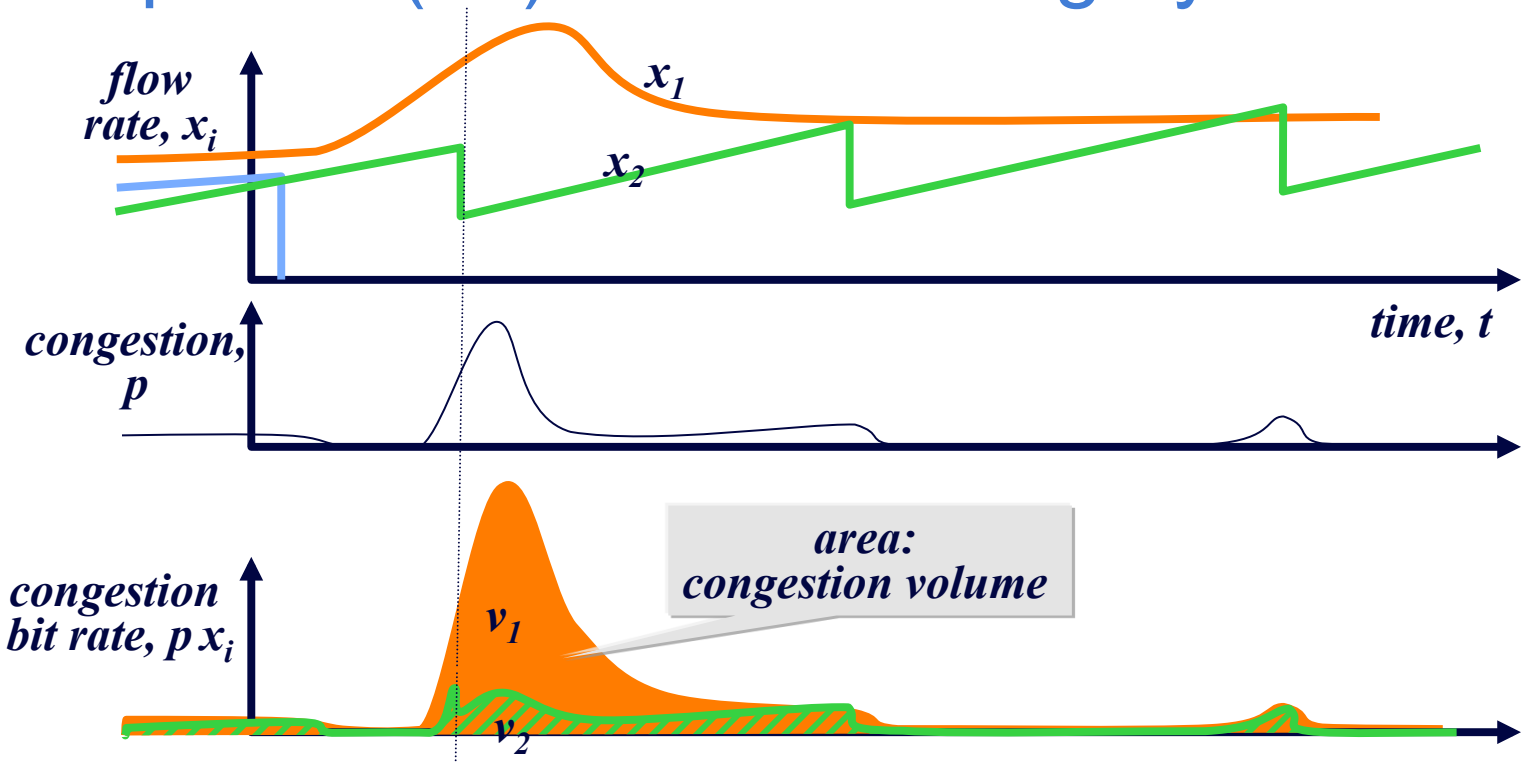

loss (marking) fraction
$p(t)$

- easy for your stack to measure
  - the amount of data discarded from your traffic
  - or marked with explicit congestion notification (ECN)
- intuition
  - some ISPs internally count volume only during peak period
  - like counting (100% x volume) during peak and (0% x volume) otherwise
  - congestion volume automagically counts (p% x volume)
- needs no wriggle-room – greatest when peaks are greatest
- makes policing of either weighted sharing approach work
  - for Diffserv down to single user granularity
  - for weighted congestion control
- counts across multiple flows and over time
  - like volume, but ignores volume in troughs
- metric for users to judge ISPs, not just ISPs to judge each customer
  - congestion is when too much traffic meets too little capacity

BT

## congestion volume

# captures (un)fairness during dynamics




- limits based on congestion volume would push back against both:
  - spiky bursts
  - traffic unresponsive to congestion

BT

# but ISPs can't limit what they can't see

- by design, transport layer handles sequence space holes
- ingress policer can't see congestion on rest of path
  - particularly not in other networks downstream
- it's not just what the Internet can do for you
  it's what you can do for the Internet
- others have independently identified this as the problem or proposed solutions
- we have a fully spec'd proposal to fix IP (re-ECN)
  - makes it in the sender's interest to reveal expected congestion in packets (packets also reveal rest-of-path congestion at borders)
  - not pushing it here (at least not today)
- focusing instead on desirable features of any solution
  - not fussed if another solution meets the same goals

# position


- quick fixes need to be compatible with long term goals
- you don't end an arms race by not working out the next move, and the next, and the next
- we have been tackling the big problem (started 8yrs ago)
  - *The great thing about the Internet is that any of the thousand million or so hosts are free to use any network equipment anywhere in the whole Internet without asking. If we're going to introduce control over what share everyone gets, how do we best preserve as much of this freedom as possible?*
- you can take our insights any way you want
  - as a benchmark to assess tactical solutions
  - as a serious protocol to be worked on
- we ask that the IETF/IRTF sets up a longer-term design team
  - to work on the general resource sharing problem?
  - to assess and guide work on quicker fixes?
  - to produce a protocol solution?

BT

thank you
please consider & argue

Q & A



# addition of re-feedback – in brief

- *before:* congested nodes mark packets
  receiver feeds back marks to sender
- *after:* sender must pre-load expected congestion
  by re-inserting feedback
- if sender understates expected compared to actual congestion, network discards packets
- result: packets will carry prediction of downstream congestion
- policer can then limit congestion caused (or base penalties on it)