Issues with LEDBAT in wide deployment

Randell Jesup

IETF 84

TSVAREA

What is LEDBAT

- draft-ietf-ledbat-congestion
  - Less-than-best-effort protocol, delay-based
  - Designed for 'background' applications
    - "Scavenger" flows
    - Avoidance of delays – helps interactivity
  - Goal to avoid interference with best-effort flows
  - Officially 'experimental', but
    - Apparently widely in-use in BitTorrent
    - Deployed in OS/X Lion for updates (confirmed)

LEDBAT details

- Approximates one-way-delay (OWD)

  - Uses approximation to target 100ms delay in bottleneck node – original number was 25ms

  - Susceptible to mis-measurement especially on new flow introduction.

  - Gets "out of the way" of the way of TCP flows on tail-drop routers

  - Not self-fair if streams use different target OWDs

Issues:

- Induces ~100ms delay on other flows across bottleneck (such as VoIP)

- Mis-measures OWD at times

- Behavior roughly similar to best-effort when faced with AQM (RED in particular was tested)

  - This means the background flow becomes foreground!

- Algorithm means it will likely out-compete any delay-based flow targeted drained buffers, such as interactive communications like adaptive VoIP and RTCWEB
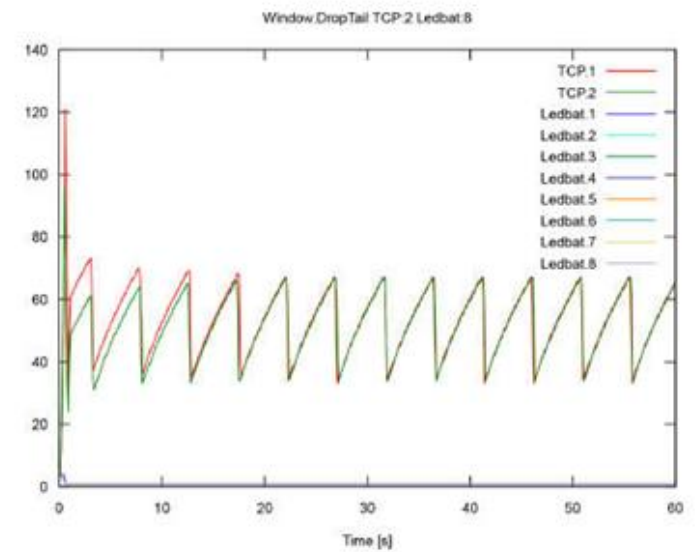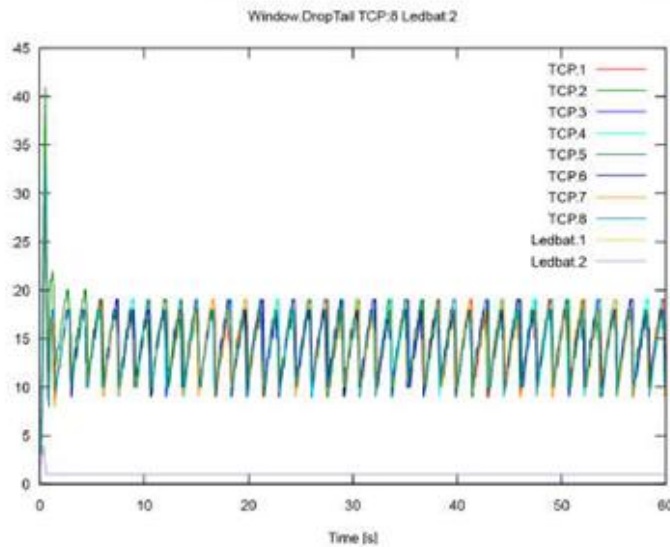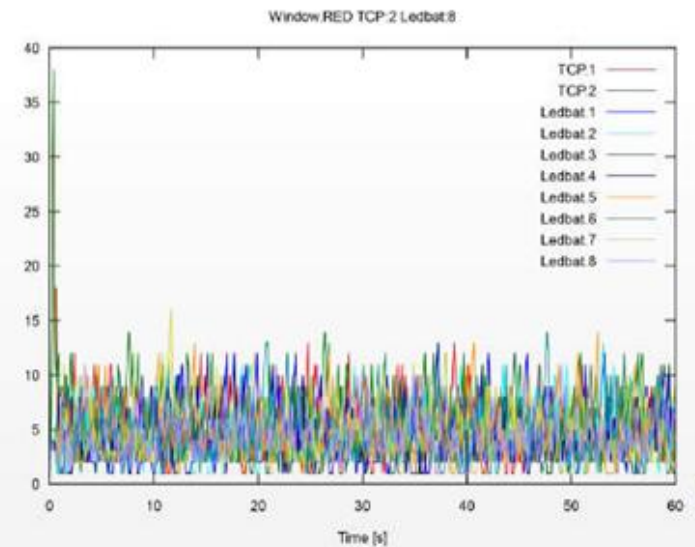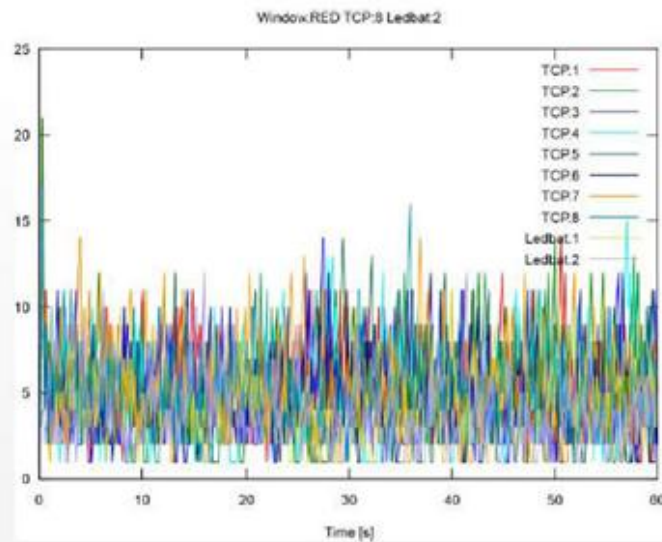
RED vs LEDBAT

VoIP and realtime flows vs LEDBAT

- Inflexible VoIP flows will experience ~100ms delays in addition to normal.

  - Mouth-to-ear target for high-quality VoIP is ~150ms, depending on echo (<100ms is better). Above 150-ms the subjective quality goes down quickly

  - With existing access-link, WiFi, codec, audio driver, and other delay, an extra 100ms will drive the flow well over 150ms total

  - Traffic-classifying home routers can mitigate this – in one direction only

- Other realtime flows (remote

VoIP vs LEDBAT
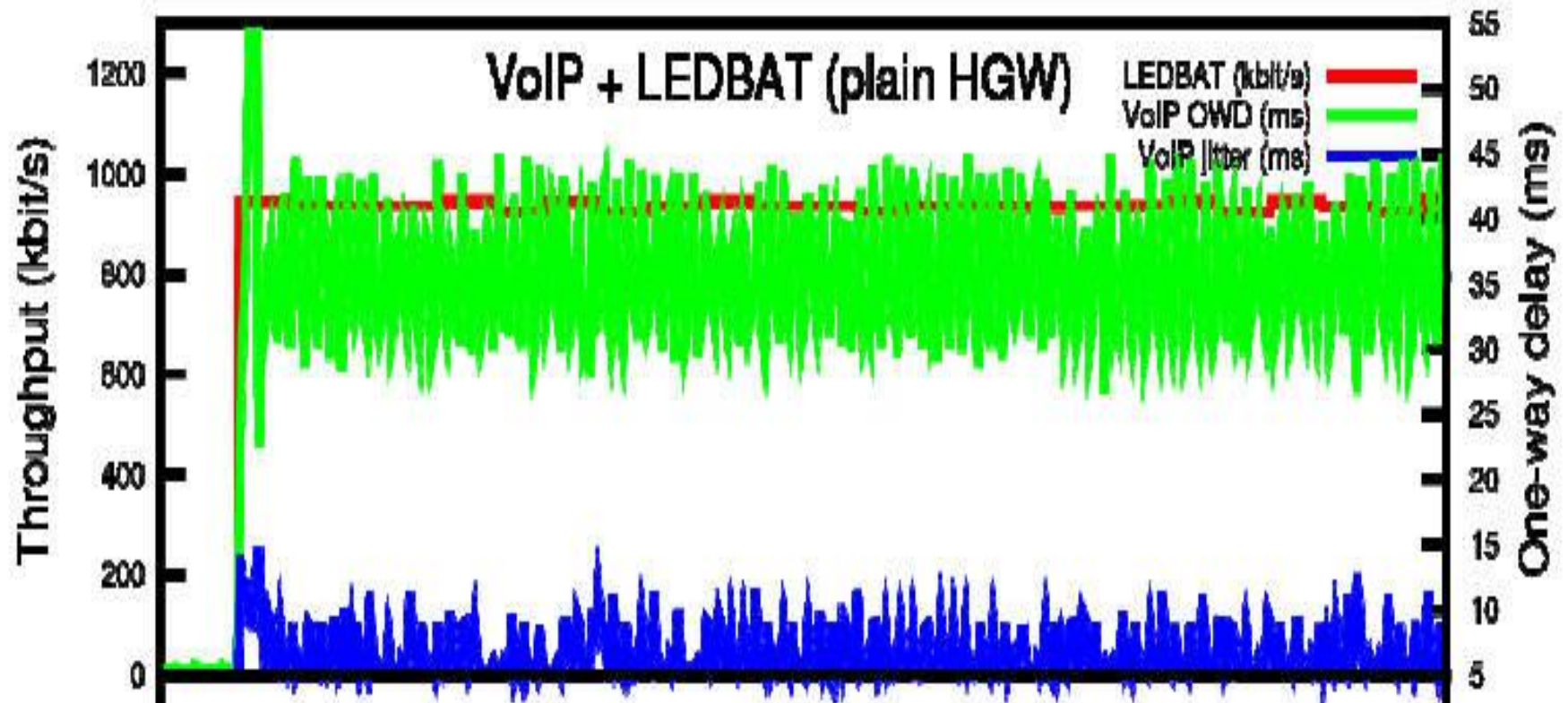
RTCWEB and adaptive flows

- Any delay-sensing algorithm such as those proposed for rtcweb, or in-use by Radvision or other proprietary applications will likely be affected by LEDBAT, and may very well end up with LEDBAT getting the majority of the bandwidth

  - This is because they typically target close-to-empty router queues, and LEDBAT is not fair with algorithms that target lower values, including tuned-lower LEDBAT flows

Suggestions for using LEDBAT

- LEDBAT isn't free. Especially problematic are cases where the user doesn't know, and can't affect, LEDBAT transfers in the background

  - Avoid it if the user doesn't know the transfer is occurring, and/or can't control its use, especially if not invoked directly by the user

  - For example, be careful using it for background machine backups, automatic syncing of datasets or images or videos, application automatic updates, etc.

  - Show some indicator that the transfer is occurring

  - Give the user some way to say "get off the

Alternatives to LEDBAT

- Investigate if LEDBAT can be modified to "play nicely" with other delay-sensing algorithms and retain it's characteristics under AQM

  - How much better does it get with 25ms instead of 100ms?

  - Incorporate some packet loss reaction for AQM

- Investigate alternative background congestion protocols, perhaps modifications to RTCWEB proposals.

  - React downwards faster/longer than "standard"?

  - Priorities?

References

Schneider, et al:

- VoIP flows with 25ms LEDBAT experience 35ms delay

- LEDBAT with even marginally higher delays (5ms) out-competes shorter-delay LEDBAT.

http://www.i-teletraffic.org/fileadmin/ITCBibDatabase/2010/schneider10.pdf

- RED causes LEDBAT to act like TCP:

http://perso.telecom-paristech.fr/~drossi/DATA/PRJ-Ledbat+AQM.pdf