To The Moon and Back
are Internet delays really that large?

Chiara Chirichella
Dario Rossi

dario.rossi@enst.fr
http://www.enst.fr/~drossi

FP7 IP project
(grant no. 318627)
Agenda

• Background
  – Bufferbloat, or the Internet is broken (again) [PAM’10, IMC’10, CACM’12, IMC’12, TMA’13, CCR’13]
  – Measuring Bufferbloat by exploiting LEDBAT [PAM’13]

• Hybrid Internet measurement campaign
  – Quantify bufferbloat (caveat of coupling bias!)
  – Root cause analysis
Background: Bufferbloat

- RTT Delay between two Internet hosts?
Background: Bufferbloat

- RTT Delay between two closeby Internet hosts?

Bufferbloat!
RTT may grow to several seconds!
Nasty impact on interactive Web, VoIP, gaming traffic, etc.

Source: [PAM’10]
Background: Bufferbloat

• Old issue, new interest
  – Previously work on buffers focused on the network core, where due to high speed, bufferbloat is limited --but situation is different at the network edge!
  – Small buffers not enough: cause of inefficiency, not fit for varying capacity (eg. WiFi) [TMA’13]
  – BitTorrent abandoned TCP in favor of LEDBAT due to this problem [PAM’10]
  – Moore law drove down memory costs, exhacerbate bufferbloat [CACM’12]

• Measurement work
  – Maximum bufferbloat via backlogged transfers [IMC’10],
  – Active testbed on 3/4G via ICMP RTT measurement [IMC’12]
  – Passive observation of TCP RTT variation on campus [CCR’13]
**Background: LEDBAT**

**TCP**

Detect congestion by losses
- Increment the congestion window \((cwnd)\) by one packet per RTT
- Halve \(cwnd\) on loss events

Consequences
- The buffer always fills up
- High delay for interactive apps
- Users need to prioritize traffic!

**LEDBAT**

Infer via delay measurement
- Senders measure minimum delay
- Queuing delay = difference wr.t. min
- React with linear controller

Aim
- At most TARGET ms of delay
- Lower priority than TCP
- Do not harm interactive application
- Avoid self-induced congestion

---

**TCP Diagram**

- \(cwnd\) increases with losses
- Time

**LEDBAT Diagram**

- \(cwnd\) approaches TARGET
- Time
Background: LEDBAT

- Exploiting LEDBAT to infer Bufferbloat [PAM’13]
  - Passive traffic observation
  - Exploit LEDBAT headers (UDP header)
  - Perform same state update sender would do
  - Queuing delay = difference with respect to minimum
  - Gauge remote peer buffer even with hidden B->C traffic
  - Very accurate vs ground truth
    - Kernel-level queue
    - Application-level RTT
Internet campaign

Hybrid methodology
• actively join torrents (M)
• passively analyze remote peers traffic (R1.. RN)

Dataset
• Nearly 100 experiments, >10 torrents (Book, Games, Software, Video, ...), 3 vantage points (FR, IT, AU)
• Available at

http://www.enst.fr/~drossi/dataset/bufferbloat-internet
Internet campaign

• Quantify bufferbloat in the wild Internet
  – Caveat: coupled sampling

• Identify root causes
  – Access type (reverse DNS)
  – Operating system (TTL fingerprint)
  – BitTorrent client (BT handshake)
Quantify bufferbloat

- Avoid bias due to coupled sampling
  - Bufferbloat samples = LEDBAT window,
  - LEDBAT window shrinks when queuing delay increases

\[ cwnd_{i+1} = \max(1, cwnd_i + \gamma(\tau - q_i)/\tau cwnd_i) \]

- Problem
  - Risk of bufferbloat understimation: few samples when the queuing delay is large
  - Risk of target delay overestimation: LEDBAT stabilizes around target delay if alone in the bottleneck

- Solution
  - Batch samples in short fixed-size windows \( t \in [t_i, t_0+W] \)
  - \( Q_i = E[q_i] \) over all packets in the window
Quantify bufferbloat

- Coupling example

Interfering TCP traffic
- Window shrinks
- Underestimate queue

LEDBAT alone
- Full window
- Overestimate target

Application limited
- Few samples, but low queuing

Transient periods
Quantify bufferbloat

- Windowed estimates ($Q_i$) correct per-packet ($q_i$) bias
- Interactivity threshold (100ms) exceeded 10% of the times
- Bufferbloat (>1s) rare, but experienced by 1% of peers
Root cause analysis

To protect herself from bufferbloat, user can:

• Change operator/access type (AT)
  – Better modem, smaller buffers, contracts with higher capacity, ...

• Change Operating System (OS)
  – Congestion control flavor of TCP interacting traffic, timestamp precision affect LEDBAT behavior, ...

• Change BitTorrent Client (BC)
  – L7 application settings: # peers, connection management policy, ...
Root cause analysis

Methodology

• Breakdown peer statistics per AT, OS and BC classes
• Quantify difference per classes
• Extract samples with equal size to avoid class imbalance
• Compare to random subset for score baselines

Metrics

• Kullback-Leibler (KL) divergence
• Hellinger Distance (HD)

Results

• Access type biggest role
• OS and BC non negligeable either

(caveat: AT, OS and BC can be correlated; e.g., all Windows peer may use uTorrent)
Conclusions

• Findings
  – Bufferbloat rare but possible (1% of peers, maybe due to concurrent BitTorrent TCP transfer to legacy peers)
  – Bufferbloat primarily affected by peer access type
  – Operating system (~TCP flavors) and application (~connection management) also correlates with queuing delay magnitude

• Limits
  – Limited to hosts running BitTorrent
  – Possibly over-estimate bufferbloat (BitTorrent)
  – Possibly spatially and temporally limited view (hosts of BT swarms, when peers use BT)
  – Data intensive (RX data from remote hosts)

• Ongoing
  – Lightweight active methodology
  – Purely passive campaign
References


[this talk] C. Chirichella, D. Rossi, To the Moon and Back: are Internet delay really that large? In IEEE INFOCOM Workshop on Traffic Measurement and Analysis (TMA'13), 2013.

Further informations on LEDBAT and bufferbloat:
http://www.enst.fr/~drossi/ledbat