Congestion Control, Active Queue Management & Bufferbloat

Olivier Mehani

Guest lecture for COMP3331/9331,
CSE, UNSW, 2014-04-16

NICTA Funding and Supporting Members and Partners

Notes

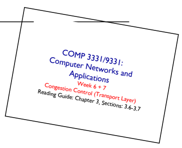
- Introduction
- Congestion Control
- Bufferbloat
- Active Queue Management
- Fighting Bufferbloat
- Conclusion

Notes

- Introduction
- Congestion Control
- Bufferbloat
- Active Queue Management
- Fighting Bufferbloat
- Conclusion

Notes

Introduction



Transport **end-to-end** congestion control adjusts throughput to fairly share available capacity[1]

Network **best effort** forwarding of all received packets

"Just works"
- ... does it?
- How much effort is best effort?
- Failure cases: Bufferbloat
  - Why does everything slow down during large file transfers?

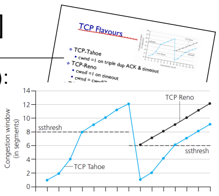
This lecture
- Congestion signals/control model
- More network involvement in throughput control
- Bufferbloat and solutions

[1] Beware of "bandwidth": radio spectrum?

Notes

Notes

TCP NewReno reminder [3]



- **Slow-start** (*cwnd* $<$ *ssthresh*): *cwnd* $\leftarrow 2 \times$ *cwnd* every RTT (or *cwnd* $+ 1$ on every ACK)
- **Congestion avoidance** (*cwnd* $>$ *ssthresh*): *cwnd* $\leftarrow$ *cwnd* $+ 1/$*cwnd* on every ACK (or *cwnd* $+ 1$ every RTT)
- Losses
  - **Fast retransmit** *cwnd*, *ssthresh* $\leftarrow$ *cwnd*$/2$ on triple dup-ACK (a.k.a. losses … or something else?)
  - **Fast recovery** *cwnd* $\leftarrow$ *ssthresh* $+ n_{\text{dupack}}$ for isolated losses
  - *cwnd* $\leftarrow 1$, *ssthresh* $\leftarrow$ *cwnd*$/2$ on timeout
- Generalised $AIMD(a, b)$ [4]
  - CA: *cwnd* $\leftarrow$ *cwnd* $+ a$
  - **Loss**: *cwnd* $\leftarrow (1 - b)$*cwnd*
  - NewReno: $a = 1, b = 1/2$

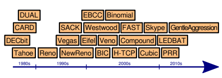
NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Congestion Control
TCP Variants

- World of TCP CC not uniform
- Default implementations in OSes vary
  - Mac OS X, FreeBSD, OpenBSD: NewReno
  - Linux: CUBIC ($\geq$ 2.6.19)
  - Windows: Compound ($\geq$Vista), Westwood+
  - Many more options in research

- Other congestion control: Vegas, HSTCP, Hybla, TFRC…
- Other transports: DCCP, SCTP, LEDBAT, …



Source: K. Winstein, N. Sivaraman, P. Thaker and H. Balakrishnan. **Transport Architectures for an Evolving Internet**. Mar. 2014. URL: http://www.ietf.org/proceedings/89/slides/slides-89-irtfopen-0.pdf

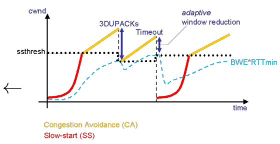
- Let's stick to TCP CC here!

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Congestion Control
TCP Variants

Westwood+ use capacity estimate on loss *cwnd*, *sshthresh* $\leftarrow$ $RTT \times BWE$



Source: http://c3lab.poliba.it/index.php/Westwood, [6]

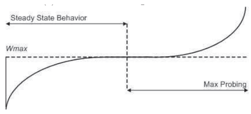
NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Congestion Control
TCP Variants

Westwood+ use capacity estimate on loss

CUBIC cubic equation flattening around last $cwnd = W_{\max}$, $cwnd \leftarrow C(t-K)^3 + W_{\max}$



S. Ha, I. Rhee and L. Xu. "CUBIC: A New TCP-friendly High-speed TCP Variant". In: **ACM SIGOPS Operating Systems Review** 42.5 (July 2008)

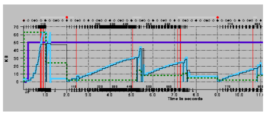
Notes

Congestion Control
TCP Variants

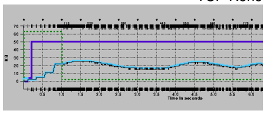
Westwood+ use capacity estimate on loss

CUBIC cubic equation flattening around last $cwnd$ $cwnd \leftarrow C(t-K)^3 + W_{\max}$

Vegas react to changes in delay



TCP Reno



TCP Vegas

Source: https://www.cs.arizona.edu/projects/protocols/

$RTT_{\text{base}} = \min(RTT)$

$Expected = cwnd / RTT_{\text{base}}$

$Diff = Expected - transferred / RTT_{\text{current}}$

$$cwnd \leftarrow \begin{cases} cwnd + 1, & \text{if } Diff < \alpha \\ cwnd - 1, & \text{if } Diff > \beta \\ cwnd, & \text{otherwise} \end{cases}$$

NICTA
Introduction
Congestion Control
TCP Variants
Bufferbloat
Active Queue Management
Fighting Bufferbloat
Conclusion

Notes

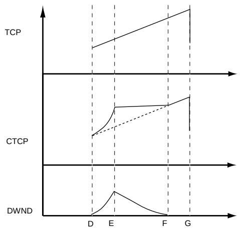
Congestion Control
TCP Variants

Westwood+ use capacity estimate on loss

CUBIC cubic equation flattening around last $cwnd$ $cwnd \leftarrow C(t-K)^3 + W_{\max}$

Vegas react to changes in delay

Compound loss-based AIMD & delay-based



K. Tan, J. Song, Q. Zhang and M. Sridharan. "A Compound TCP Approach for High-Speed and Long Distance Networks". In: **INFOCOM 2006**. Apr. 2006

$cwnd \leftarrow cwnd + dwnd$

$$dwnd \leftarrow \begin{cases} dwnd + \begin{cases} \alpha dwnd^k - 1, & \text{if } Diff < \gamma \\ \eta Diff, & \text{if } Diff \geq \gamma \end{cases} \\ dwnd(1-\beta), & \text{on loss } (\ldots\text{MD}) \end{cases}$$

Vegas-ish "AI. . . "

NICTA
Introduction
Congestion Control
TCP Variants
Bufferbloat
Active Queue Management
Fighting Bufferbloat
Conclusion

Notes

Congestion Control
TCP Variants

Westwood+ use capacity estimate on loss

CUBIC cubic equation flattening around last $cwnd$ $cwnd \leftarrow C(t-K)^3 + W_{\max}$

Vegas react to changes in delay

Compound loss-based AIMD & delay-based

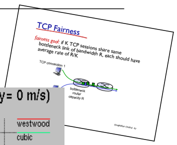
- Three main CC classes
  - loss-based
    - $AIMD(a, b), \forall a, b$
    - quadratic, cubic laws instead of AI. . .
    - estimates instead of . . . MD
  - delay-based
    - compare actual throughputs (ACKs) to expected ($cwnd/RTT$)
    - may reduce *cwnd* without loss
  - hybrids
    - kitchen sink

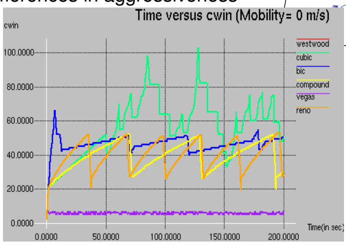
NICTA
Introduction
Congestion Control
TCP Variants
Bufferbloat
Active Queue Management
Fighting Bufferbloat
Conclusion

Notes

Congestion Control
TCP Variants

- Not all CC are equal: fairness
- Differences in aggressiveness





Source: M. Jehan, G. Radhamani and T. Kalakumari. "VEGAS: Better Performance Than Other TCP CongestionControl Algorithms on MANETs". In: **International Journal of Computer Networks** 3.2 (May 2011)

Notes

Congestion Control
TCP Variants

- Bottomline: fill the pipe as much as possible
  - *i.e.*, put as many packets in flight as the network can buffer
  - until a loss occurs
  - except for Vegas (delay-based)
- Sounds reasonable: make sure the resource is used fully
- Worked since the 90s

Notes

- Introduction
- Congestion Control
- Bufferbloat
  - Router design 90s–2011
  - Too much buffering is not necessarily best
  - When does it happen?
- Active Queue Management
- Fighting Bufferbloat
- Conclusion

Notes

Bufferbloat
Router design 90s–2011[2]

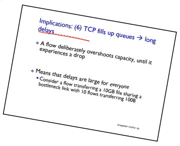
- Best-effort
- Losses are bad
- Buffers avoid losses during bursts
- Memory is cheap
- Increase buffer size so losses never happen
- Rule-of-thumb: Bandwidth × Delay Product (BDP)

[2] This is slightly caricatured

Notes

Bufferbloat
Router design 90s–2011

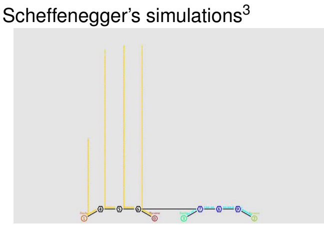
- Are losses really that bad?
- Could they be useful for something?
- How does loss-based CC behave without losses?



Notes

Bufferbloat
Too much buffering is not necessarily best

- Heavy download vs. interactive traffic
- Richard Scheffenegger's simulations[3]



Left: large buffers; Right: small buffers

  - CC doesn't get needed feedback until much later
- Delay (from BDP) at **every** hop

[3] https://lists.bufferbloat.net/pipermail/bloat/2011-February/000082.html

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Bufferbloat[5]
When does it happen?

- Traffic sent from high-speed link to lower-speed link
  - Ethernet to Wi-Fi
  - Ethernet to ADSL
  - Fibre to Ethernet
- Oversized buffers are everywhere
  - Routers
  - Switches
  - Netwok interfaces
  - Network interface drivers
    - Patches for some Linux drivers
- Some CC unaffected: Vegas, LEDBAT[4]
  - Delay-based CC manages to avoid delay (shock!)
  - Get killed by the others, more aggressive, CC

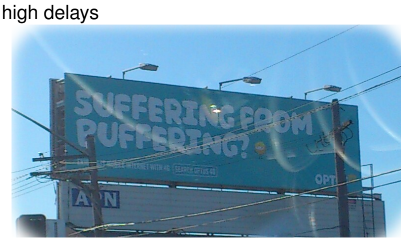
[4] Low Extra Delay BAckground Transport
[5] http://www.bufferbloat.net/

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Bufferbloat

- Aggressive CC filling the pipes
- Buffers too large
- High throughput (eventually)
- But high delays



Source: myself (:

- Note: some buffering is good (*e.g.*, to absorb temporary bursts)
- Solutions?
  - Deployable?

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

- Introduction
- Congestion Control
- Bufferbloat
- Active Queue Management
  - Techniques and Objectives
  - AQM variants
- Fighting Bufferbloat
- Conclusion

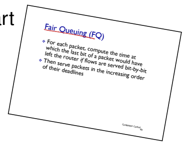
Notes

Active Queue Management

Different ways to control traffic rates

Congestion Control smart endpoints, dumb routers (with lots of memory!)

Packet-scheduling (*e.g.*, FQ) very smart routers which manage multiple queues, classify traffic, and distribute it



Active Queue Management smart-ish routers which don't always forward packets

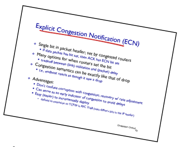
NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Active Queue Management

Techniques and Objectives

- Simple per-packet rules
  - Selective drops before queues are full
  - Packet-marking (*e.g.*, ECN)



- Variable goals
  - Fairly share limited network capacity
  - Keep queues short (longer $\simeq$ congestion)
  - Limit delays (mitigate Bufferbloat!)

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Active Queue Management

AQM variants

RED (Random Early Detection [5]) randomly drops/marks **arriving packets** depending on **current queue size**
- Initally hard to configure
- Well known [1]
- Adaptive RED (ARED [2]) automatically adjust most parameters

CoDel (Controlled Delay [10]) drops/marks **outgoing packets** depending on their **sojourn time** in the queue
- Designed for Bufferbloat
- Hardcoded target delay: 5 ms

PIE (Proportional Integral controller Enhanced [11]) randomly drops/marks **arriving packets** depending on their **queueing latency**

***-CoDel** Fair Queueing, SFQ, ...

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Notes

Fighting Bufferbloat

Impact of various CC and various AQMs[6]

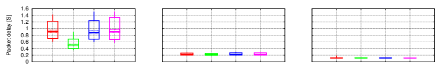
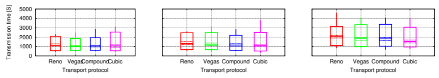


Figure 5: Packet transmission times



(a) DropTail (b) RED (c) CoDel

Figure 6: Time needed to transmit 10 MB

- Least delay introduced by Vegas
- Most delay introduced by CUBIC
- AQMs such as RED and CoDel provide an important reduction in delay
- RED allows to achieve a slightly better throughput

[6] N. Kuhn, E. Lochin and O. Mehani. "Revisiting Old Friends: Is CoDel Really Achieving What RED Cannot?" In: Under review

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

- Mainly loss-based congestion control
  - Should we avoid losses...
  - ...or a new congestion collapse?
- Too big buffers lead to Bufferbloat

[7]

  - Rules-of-thumb ($C \times D$) sometimes smash the thumb
  - More is not always better
- Use smarter queue management (*e.g.*, ARED)
- Add some delay dependence

  Layer 4 Vegas, Compound, LEDBAT (w/o AQM!)
  Layer 3-ish Active Queue Management: CoDel, PIE
  - CeroWRT firmware with CoDel variants[8]
- **Consider the whole system you are working with**

Thanks. Questions?

<olivier.mehani@nicta.com.au>

[7] Source: http://www.fg-a.com/clipart.html
[8] http://www.bufferbloat.net/projects/cerowrt

NICTA
Introduction
Congestion Control
TCP NewReno reminder
TCP Variants
Bufferbloat
Router design 90s–2011
Too much buffering is not necessarily best
When does it happen?
Active Queue Management
Techniques and Objectives
AQM variants
Fighting Bufferbloat
Conclusion

Notes

Opportunities at Nicta
Come study with us!

- Various options with the Network Research Group
  - Internships
  - Projects (*e.g.*, Hons.)
  - PhD (w/ UNSW)
- Australian Technology Park, near Redfern
- http://nrg.nicta.com.au/internships/

Notes

Backup

Notes

S. Floyd. **RED: Discussions of Setting Parameters**. Nov. 1997. URL: http://www.icir.org/floyd/REDparameters.txt.

S. Floyd, R. Gummadi and S. Shenker. **Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management**. Tech. rep. Aug. 2001.

S. Floyd, A. Gurtov, Y. Nishida and T. Henderson. **The NewReno Modification to TCP's Fast Recovery Algorithm**. RFC 6582. Apr. 2012.

S. Floyd, M. Handley and J. Padhye. **A Comparison of Equation-Based and AIMD Congestion Control**. Tech. rep. May 2000.

S. Floyd and V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance". In: **IEEE/ACM Transactions on Networking** 1.4 (Aug. 1993).

Notes

L. Grieco and S. Mascolo. "TCP Westwood and Easy RED to Improve Fairness in High-Speed Networks". In: **Protocols for High Speed Networks**. Vol. 2334. Lecture Notes in Computer Science. May 2002. Chap. 9.

S. Ha, I. Rhee and L. Xu. "CUBIC: A New TCP-friendly High-speed TCP Variant". In: **ACM SIGOPS Operating Systems Review** 42.5 (July 2008).

M. Jehan, G. Radhamani and T. Kalakumari. "VEGAS: Better Performance Than Other TCP CongestionControl Algorithms on MANETs". In: **International Journal of Computer Networks** 3.2 (May 2011).

N. Kuhn, E. Lochin and O. Mehani. "Revisiting Old Friends: Is CoDel Really Achieving What RED Cannot?" In: Under review.

K. Nichols and V. Jacobson. "Controlling Queue Delay". In: **Queue** 10.5 (May 2012).

Notes

- R. Pan et al. **PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem**. Internet-Draft draft-pan-aqm-pie-00.txt. June 2013.
- K. Tan, J. Song, Q. Zhang and M. Sridharan. “A Compound TCP Approach for High-Speed and Long Distance Networks”. In: **INFOCOM 2006**. Apr. 2006.
- K. Winstein, N. Sivaraman, P. Thaker and H. Balakrishnan. **Transport Architectures for an Evolving Internet**. Mar. 2014. URL: http://www.ietf.org/proceedings/89/slides/slides-89-irtfopen-0.pdf.

NICTA

Backup

References

Notes

Notes

Notes

Notes