CoDel++: Improving the Delay-Throughput Trade-off

<Internet Draft - 00 is in progress>

Mohit P. Tahiliani

HotRFC Lightning Talks

IETF 123

Credits:

Deepa Kumari, Vivek Jain, Aniket Singh, Arun Kumar Rajaraman, Ayush Nigam,

Monika Gautam, Jayesh Akot

Working of CoDel (RFC 8289)

- Two phases and two configurable parameters in CoDel algorithm
  - Dropping phase and Non-dropping phase (default initial phase)
  - Configurable parameters: target (default: 5 ms) and interval (default: 100ms)
- Non-dropping phase:
  - Calculate current queue delay (cur_qdelay) for every outgoing packet:

    cur_qdelay = dequeue_time - enqueue_time Eq. (1)
  - When cur_qdelay > target
    - Start the ‘interval’ timer (100 ms)
  - Enter dropping phase, only if cur_qdelay > target for the entire ‘interval’
  - Otherwise, stay in non-dropping phase because it was a short burst.
    - Reset ‘interval’ timer and wait for cur_qdelay > target again

Working of CoDel (contd …)

- On entering the dropping phase
  - Drop the outgoing packet
  - Increment ‘count’ (number of packets dropped in ‘this’ dropping phase)
  - Calculate time to drop the next packet using the Control Law:

$$next_drop_time = cur_time + \frac{interval}{\sqrt{count}}$$

Eq. (2)

assumes Reno CCA

- While in dropping phase
  - Continue to drop packets using Eq. (2) till cur_qdelay > target
  - When cur_qdelay < target, exit the dropping phase
  - Update the value of count as mentioned in RFC 8289
  - Enter non-dropping phase

Working of CoDel (contd …)

- Example: CoDel's working in the dropping phase

  Assume:

  - current_time = 1000ms and count = 1, then

    next_drop_time = 1000ms + 100ms/sqrt(1)

    next_drop_time = 1100ms

  - Assume that the second packet is dropped at 1100ms (current time), then

    next_drop_time = 1100ms + 100ms/sqrt(2)

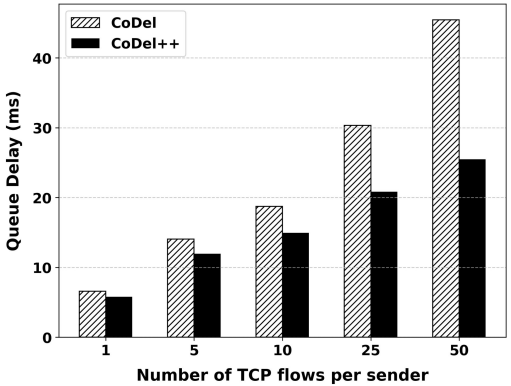
    next_drop_time = 1100ms + 71ms = 1171ms 4

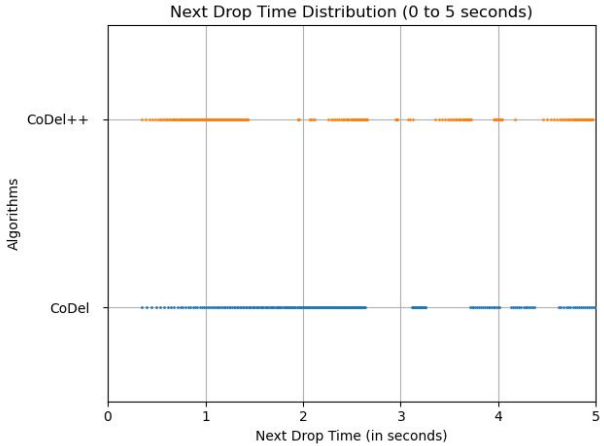
  - The gap between packet drops reduces if CoDel stays longer in dropping phase:

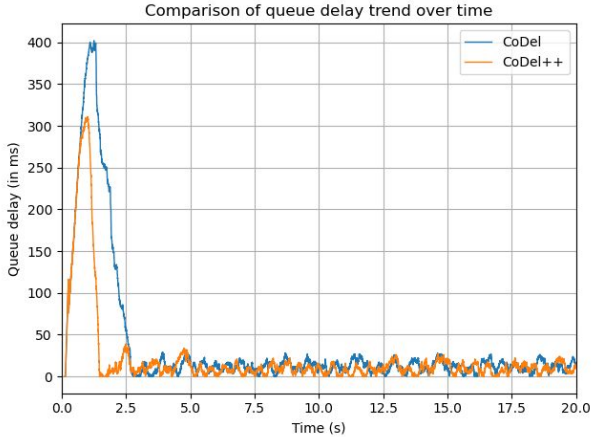
    Drop times in this example: 1000, 1100 (1000 + 100), 1171 (1100 + 71), 1228 (1171 + 57), 1278 (1228 + 50), …

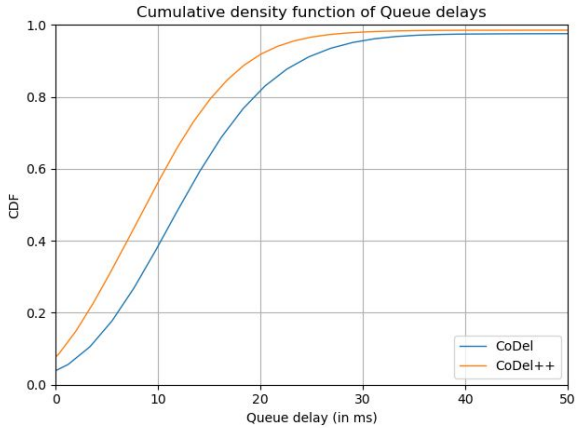
CoDel++

- Uses a new control law!
  - No other change in RFC 8289
  - Linux and ns-3 implementations available for testing
- Looking for collaborators who can help with testing in real deployments









Thank you!

Mohit P. Tahiliani
National Institute of Technology Karnataka, India
tahiliani@nitk.edu.in