**Single Queue CoDel Analysis**

**Anil Agarwal**

**ViaSat Inc.**

**Jun 16, 2015**

1. **CoDel Packet Drop Rate Analysis**

Under sustained congestion, every second, the CoDel packet drop rate increases approximately by DRI packets/second.

DRI = 0.5 / I\_init^2

Define I(n) as the drop interval size after n consecutive congested intervals.

Define DR(n) as the packet drop rate in the nth cycle

Define DRI as the packet drop rate increase per second

I(n) = I\_init / sqrt(n)

DR(n) = 1 / I(n)

I(n+1) = I\_init / sqrt(n+1)

DR(n+1) = 1 / I(n+1)

DRI = (DR(n+1) - DR(n)) / I(n)

DRI = (sqrt(n+1) / I\_init - sqrt(n) / I\_init) / (I\_init / sqrt(n))

DRI = 1 / I\_init^2 \* (sqrt(n+1) - sqrt(n)) \* sqrt(n)

DRI = 1 / I\_init^2 \* (sqrt(n^2+n) - n)

Let S = sqrt(n^2+n) - n

Using the equation a^2 - b^2 = (a + b) \* (a - b)

i.e., a - b = (a^2 - b^2) / (a + b)

and a = sqrt(n^2+n), b = n

S = (n^2 + n - n^2) / (sqrt(n^2+n) + n)

S = n / (sqrt(n^2+n) + n)

S = 1 / (sqrt(n^2+n)/n + n/n)

S = 1 / (sqrt(1+1/n) + 1)

S ~= 1 / (sqrt(1) + 1) for large n

S ~= 0.5

**DRI = 0.5 / I\_init^2 (1)**

DRI does not depend on link capacity or rtt.

Note that the drop rate will stabilize when it reaches the right value for the given traffic.

Examples:

I\_init = 100 ms, DRI = 50 packets/sec/sec

I\_init = 50 ms, DRI = 200 packets/sec/sec

I\_init = 500 ms, DRI = 2 packets/sec/sec

1. **Analysis of CoDel Convergence Time for Large Number of TCP Connections**

Given:

Link Capacity = C bps

Number of active TCP (or TCP-like) connections = n

Round Trip Time = rtt seconds

Packet size = PS bytes

CoDel initial interval = I\_init seconds

Assume:

All TCP connections are long-lived.

At steady state, for every TCP conn, cwnd increases linearly from W/ 2 to W

by one packet per RTT and then drops to W / 2 when a packet is dropped or

ECN-marked

Derive:

bps\_per\_conn = C / n

Conn max window size = W = bps\_per\_conn \* rtt / PS (packets)

Ideal drop interval per conn = DI1 = W / 2 \* rtt (seconds)

Ideal drop rate per conn = DR1 = 1 / DI1 (packets/sec)

Aggregate Ideal\_drop\_rate = DR = DR1 \* n (packets/sec)

CoDel time to reach ideal drop rate = T = DR / DRI (seconds)

or

**CoDel time to reach ideal drop rate = T = 4 \* PS / C \* (n \* I\_init / rtt) ^ 2 (seconds) (2)**

Comments

The time T taken by CoDel to reach ideal drop rate is -

- Inversely proportional to link Capacity C

- Proportional to the square of the number of active TCP connections

- Proportional to the square of the initial interval value

- Inversely proportional to the square of rtt.

If link capacity increases and the number of TCP connections increases

correspondingly, then T will increase.

During time 0 to T, the queue size will be high and packets may experience tail-drops.

In steady state, large sudden changes in the number of active connections will likely result in tail-drops.

This analysis is probably not applicable when rtt > I\_init or if W is too small.

Numerical Examples

1. C = 1 Gbps, n = 1000, rtt = 100 ms, PS = 1500 bytes, I\_init = 100 ms

DRI = 50 packets/sec/sec

W = 8.33 packets

Drop Rate DR = 2400 packets/sec

T = 48 seconds

2. C = 10 Gbps, n = 10000, rtt = 100 ms, PS = 1500 bytes, I\_init = 100 ms

DRI = 50 packets/sec/sec

W = 8.33 packets

Drop Rate DR = 24000 packets/sec

T = 480 seconds

This does seem to indicate that for core and edge routers that handle large number of connections, if the number of connections changes by large amounts, then the convergence of the CoDel Interval value will likely take a correspondingly large amount of time.

fq\_codel helps reduce the magnitude of the problem, since the number of connections per queue is smaller, but can still be fairly large for core routers.

This issue is also present with unresponsive UDP flows.

1. **CoDel Count Value Analysis**

During sustained congestion, CoDel count value increases with time as –

C(t2) = (sqrt(C(t1)) + 0.5 \* (t2 – t1) / I\_init)^2

Or approximately as –

C(t2) = C(t1) + (t2 – t1) / I(t1)

where I(t1) is the drop interval size at time t1.

Define C(t) as the count value at time t.

Define DR(t) as the packet drop rate at time t

Define DRI as the packet drop rate increase per second

I(t1) = I\_init / sqrt(C1),

DR(t1) = 1 / I(t1)

I(t2) = I\_init / sqrt(C2),

DR(t2) = 1 / I(t2)

DR(t2) = DR(t1) + DRI \* (t2 - t1)

sqrt(C2) / I\_init = sqrt(C1) / I\_init + DRI \* delta\_t

where delta\_t = t2 – t1

Substituting DRI = 0.5 / I\_init^2

sqrt(C2) / I\_init = sqrt(C1) / I\_init + 0.5 / I\_init^2 \* delta\_t

sqrt(C2) = sqrt(C1) + 0.5 \* delta\_t / I\_init

**C2 = (sqrt(C1) + 0.5 \* delta\_t / I\_init) ^ 2 (3)**

C2 = C1 + sqrt(C1) \* delta\_t / I\_init + 0.25 \* delta\_t^2 / I\_init^2

C2 = C1 + sqrt(C1) \* delta\_t / I\_init + 0.25 \* delta\_t^2 / I\_init^2

C2 = C1 + delta\_t / I(t1) + 0.25 \* delta\_t^2 / I\_init^2

This can be approximated for small delta\_t by dropping the second term -

**C2 = C1 + delta\_t / I(t1) (4)**

1. **Limit on value of count variable**

The count value must not be allowed to increase in an unbound manner or to roll-over.

The drop interval must not be allowed to decrease to a 0 value.

Given,

I(n) = I\_init / sqrt(count)

Assume I\_init is maintained as an integer value.

Then, to keep I(count) >= 1, the following condition should be met –

I\_init / sqrt(count) >= 1

sqrt(count) <= I\_init

**count <= I\_init^2** (5)

For example, for I\_init = 100,000 microseconds, count must be <= 1E10.

For a 32-bit representation of count, this is higher than the maximum 32-bit value.

Hence, the limit should be 2^32-1 and the count variable should not be allowed to roll-over.

For I\_init = 50,000 microseconds, count must be <= 2.5E9.

This limit is not reached under normal conditions but may be reached with high-rate unresponsive UDP traffic.