## Current Implementation

Assumption: Unless explicitly mentioned that some condition is not satisfied, all necessary conditions for marking packets are satisfied

- ECT1 and ECT0 packets are treated similarly
- If delay is more than target delay then instead of dropping, packets are marked with the reason “TARGET_EXCEEDED_MARK”.
- if the delay is greater than CE threshold then packets are marked with reason “CE_THRESHOLD_EXCEEDED_MARK”
- IF CE threshold is less than the target delay then both the reasons for marking are logged.

## Proposal

**Implementation available at:**
**https://gitlab.com/bhaskar792/ns-3-dev/-/commits/L4S-fqcodel**

Assumption: Unless explicitly mentioned that some condition is not satisfied, all necessary conditions for marking packets are satisfied

- A boolean attribute will be added for L4S in fq-codel-queue-disc.cc and only when the attribute is enabled, the behavior will change as described in points below.
- If L4S attribute is enabled then it will be mandatory to set CE threshold
- ECT1 traffic will only be marked at CE threshold (Target delay doesn’t matter)
- ECT0 packets will be marked if queue delay is greater than target delay( CE threshold doesn’t matter)
- If L4S is enabled then ECN must be enabled
- If L4S is enabled then ECT1 traffic will bypass the entire codel process

If there should be some changes in the proposed implementation **please comment** here. Detailed implementation will be added to this doc after confirming that the proposed approach is fine.

### UseEcn vs UseL4s

- UseEcn = True and UseL4s = false then current implementation
- UseEcn = True and UseL4s = True then only ECT1 packets are marked at CE threshold
- UseEcn = False and UseL4s = True then raise a fatal error

Basic Implementation in fq-codel-queue-disc.cc

```
if (codel)
        {
```

```
          // If L4S is enabled then enable ECN too
          codel->SetAttribute ("UseEcn", BooleanValue (m_useEcn));
          codel->SetAttribute ("CeThreshold", TimeValue
(m_ceThreshold));
          codel->SetAttribute ("UseL4s", BooleanValue (m_useL4s));
        }
```

Changes in codel-queue-disc.cc

Implementation is available at https://gitlab.com/bhaskar792/ns-3-dev/-/commits/L4S-fqcodel

Key points

- Attribute useL4s is added which uses m_useL4s in both CoDel and FqCoDel
- If L4S is enabled then ECT1 packets will bypass the whole CoDel process

## Test Cases

1. Without hash collisions
    - There will be 2 queues say q1 and q2
    - Enqueue 70 ECT0 packets in q1 and 70 ECT1 packets in q2
    - Enqueue both ECT0 and ECT1 packet in their separate queues with a delay of 0.5ms, such that at 0.5 ms there will be 1 ECT0 packet in q1 and 1 ECT1packet in q2
    - Dequeue packets from the queues with the delay of 1ms

      | Time | nEnq/nDeqPackets in Q1 | nEnq/DeqPackets in Q2 |
      |---|---|---|
      | 0 | 0/0 | 0/0 |
      | 0.5 | 1/0 | 1/0 |
      | 1.0 | 2/1 | 2/0 |
      | 1.5 | 3/1 | 3/0 |
      | 2.0 | 4/2 | 4/0 |

      nEnq/nDeq = number of enqueued packets/ number of dequeued packets

    - At 5ms, the queue delay will be 2.5 for q1 so the queue having ECT1 packets will start marking the packets **considering CE threshold to be 2ms**.
    - At 140 ms(last packets will be dequeued at 140ms), there should be 66 CE threshold marks as all the packets after the 4th packet will be marked (5th packet

should have a delay of 2.5ms) and 1 target exceeded mark as the last dequeue will be at 140ms and all the packets dequeued after 10ms will have a  delay of more than 5ms (target delay) and as the interval is 100ms so there should be exactly 1 target exceeded mark at 140ms

2. With hash collisions
    - There will be only 1 queue
    - Enqueue 70 ECT0 packets and 70 ECT1 packets
    - Enqueue 1 ECT1 packet at 0.5ms then Enqueue 1 ECT0 packet at 1ms and so on, so enqueueing with the delay of 0.5ms, and dequeue 1 packet at 1ms then at 2ms so dequeueing with a delay of 1ms.

| Time | ECT1 nEnq/nDeqPackets | ECT0 nEnq/nDeqPackets |
|---|---|---|
| 0 | 0/0 | 0/0 |
| 0.5 | 1/0 | 0/0 |
| 1.0 | 1/1 | 1/0 |
| 1.5 | 2/1 | 1/0 |
| 2.0 | 2/2 | 2/0 |
| 2.5 | 3/2 | 2/0 |
| 3.0 | 3/3 | 3/0 |

    - At 5ms, the queue delay for ECT1 packet enqueued at 2.5ms will be 2.5ms so the ECT1 packets will start getting marked considering CE threshold to be 2ms
    - At 140ms, there should be 68 CE threshold marks and 1 target exceeded mark