8 Congestion Control

NFD now implements a simplified version of the congestion control scheme in "A Practical Congestion Control Scheme for NDN" [22]. Our approach consists of three parts:

1. **Congestion Detection:** Each node detects congestion locally by monitoring its outgoing queues.
2. **Congestion Signaling:** After detecting congestion, NFD marks Data packets to inform consumers (and potentially downstream routers).
3. **Consumer Rate Adjustment:** End consumers react to congestion signals by adjusting their Interest sending rate.

8.1 Congestion Detection

Every NFD face monitors its *outgoing queue size* to decide whether the link is congested. More specifically, a face applies a simplified version of the CoDel Active Queue Management scheme [23].

This scheme uses an marking interval (default: 100ms) and a congestion threshold (default: 64KB). A face enters the congested state (and marks the first packet) when the queue size first exceeds the threshold. As long as the queue size stays above the threshold, the marking interval will be iteratively decreased. Once the queue size falls below the threshold, the faces leaves the congested state and the marking interval is reset to the default value.

The congestion detection can be enabled in multiple ways:

- The default is configured in the NFD configuration file by setting `face_system.general.enable_congestion_marking` to `yes` or `no`
- For a specific face, congestion detection can be enabled/disabled through `nfdc face create/update` by using the `congestion-marking on/off` parameter. This will override the value set in the NFD configuration file.

Congestion detection is currently not supported on websockets.

8.2 Congestion Signaling

Once a face enters the *congested state*, congestion is signaled with a single bit in the NDNLPv2 header[17].

This congestion bit can be read and written by both consumers and producer applications:

- `uint64_t PacketBase::getCongestionMark()`
- `void PacketBase::setCongestionMark(uint64_t mark)`

Where PacketBase means Interest, Data, or NACK, and a mark value of 0 indicates that the mark is unset.

8.3 Consumer Adaptation

The consumer can use congestion marks as an early indication of congestion, and thus reduce its sending rate before the queue buffer overflows. Hence, a consumer should treat congestion marks just like timeouts (trigger a window decrease). The CoDel scheme ensures that congestion marks are infrequent (starts with 1 per 100ms), unless the congestion reaches extreme levels.

Currently, the application `ndncatchunks` supports congestion reaction by default. For testing, it can be disabled with the command `--aimd-ignore-cong-marks`

[17] The header format is defined as "nonNegativeInteger" in order to allow extensions of more fine-grained congestion signaling.