Changing Ack Frequency

draft-iyengar-quic-delayed-ack
https://janaiyengar.github.io/ack-frequency

QUIC WG, !Prague, March 2021
Background

RFC 5681 recommends ACK every 2 packets for TCP

In practice, ACK collapsing (thinning) is widespread for TCP
  at endhosts
  by middleboxes

These optimizations are critical for
  high bandwidth links
  highly asymmetric links (satellite)
**Background**

QUIC transport currently recommends the same as TCP

Sending acks is expensive (CPU)

data receiver prefers to send fewer acks

(see [Issue 3304](#), [Issue 1978](#) for discussion)
Background

QUIC transport currently recommends the same as TCP

Sending acks is expensive (CPU)
  data receiver prefers to send fewer acks
  (see [Issue 3304](https://github.com/quicwg-core/issues/issues/3304), [Issue 1978](https://github.com/quicwg-core/issues/issues/1978) for discussion)

Fewer acks can cause poor performance
  window-based CC (Reno, Cubic) is driven by ack events
  delaying acks decreases throughput of these controllers
  data sender knows tolerance
Background

QUIC transport currently recommends the same as TCP

Sending acks is expensive (CPU)
  data receiver prefers to send fewer acks
  (see Issue 3304, Issue 1978 for discussion)

Fewer acks can cause poor performance
  window-based CC (Reno, Cubic) is driven by ack events
  delaying acks decreases throughput of these controllers
  data sender knows tolerance

Data sender may want to *increase* ACK rate
  new startup schemes (eg, paced chirping)
Summary of Incentives

Data receiver:
  wants to send fewer acks

Data sender:
  knows tolerance
  wants to control ack rate
Proposal

Data receiver is naturally incentivized to ack minimally

Need to communicate data sender’s desire/tolerance

Solution:
Frame from data sender to change
data receiver’s ack behavior
ACK_FREQUENCY frame

0xAF (i) ...

Sequence Number (i) ...

Packet Tolerance (i) ...

Update Max Ack Delay (i) ...

Ignore Order (8) |

0xAF : Frame Type
# ACK_FREQUENCY frame

<table>
<thead>
<tr>
<th>Sequence Number (i)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Tolerance (i)</td>
<td>...</td>
</tr>
<tr>
<td>Update Max Ack Delay (i)</td>
<td>...</td>
</tr>
<tr>
<td>Ignore Order (8)</td>
<td></td>
</tr>
</tbody>
</table>

**Sequence Number:**  
Ensures consistent processing order
### ACK_FREQUENCY frame

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
+-----------------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       0xAF (i)                         …               |
+-----------------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number (i)                    …          |
+-----------------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Packet Tolerance (i)                 …         |
+-----------------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Update Max Ack Delay (i)               …     |
+-----------------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Ignore Order (8) |                                |
```

**Packet Tolerance:**

Number of ack-eliciting packets before an immediate ACK

Changes default of 2 to be a peer-controlled variable
**ACK_FREQUENCY frame**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

`0xAF (i)` ...

`Sequence Number (i)` ...

`Packet Tolerance (i)` ...

`Update Max Ack Delay (i)` ...

`Ignore Order (8)`

---

**Update Max Ack Delay:**

Updates receiver’s max_ack_delay in microseconds
Changes max_ack_delay to be a peer-controlled variable
How low can “Update Max Ack Delay” be?

Transport Parameter: min_ack_delay (0xff02de1a)

the minimum amount of time (in microseconds) by which the endpoint can delay an acknowledgement

Used for negotiating use of this extension
ACK_FREQUENCY frame

Ignore Order:

0x01 means *always* delay (even on reordering)
Used by data senders that expect or observe reordering
Status

We have to iron some issues out (open on github) authors have been busy with core drafts might be good to get more wg input at this point

Propose adopting as wg item