QUIC Packet Format

Long header

Short header
QUIC Packet Format

Long header

Short header
QUIC Packet Format

Long header

Short header
Frames

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Frame 1 (*)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Frame 2 (*)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Frame N (*)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| ...
### Frames

<table>
<thead>
<tr>
<th>Frame Type (i)</th>
<th>Type-Dependent Fields (*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame 1 (*)</td>
<td>...</td>
</tr>
<tr>
<td>Frame 2 (*)</td>
<td>...</td>
</tr>
<tr>
<td>Frame N (*)</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Frame Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>PADDING</td>
<td></td>
</tr>
<tr>
<td>0x01</td>
<td>PING</td>
<td></td>
</tr>
<tr>
<td>0x02 - 0x03</td>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td>RESET_STREAM</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>STOP_SENDING</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>CRYPTO</td>
<td></td>
</tr>
<tr>
<td>0x07</td>
<td>NEW_TOKEN</td>
<td></td>
</tr>
<tr>
<td>0x08 - 0x0f</td>
<td>STREAM</td>
<td></td>
</tr>
<tr>
<td>0x10</td>
<td>MAX_DATA</td>
<td></td>
</tr>
<tr>
<td>0x11</td>
<td>MAX_STREAM_DATA</td>
<td></td>
</tr>
<tr>
<td>0x12 - 0x13</td>
<td>MAX_STREAMS</td>
<td></td>
</tr>
<tr>
<td>0x14</td>
<td>DATA_BLOCKED</td>
<td></td>
</tr>
<tr>
<td>0x15</td>
<td>STREAM_DATA_BLOCKED</td>
<td></td>
</tr>
<tr>
<td>0x16 - 0x17</td>
<td>STREAMS_BLOCKED</td>
<td></td>
</tr>
<tr>
<td>0x18</td>
<td>NEW_CONNECTION_ID</td>
<td></td>
</tr>
<tr>
<td>0x19</td>
<td>RETIRE_CONNECTION_ID</td>
<td></td>
</tr>
<tr>
<td>0x1a</td>
<td>PATH_CHALLENGE</td>
<td></td>
</tr>
<tr>
<td>0x1b</td>
<td>PATH_RESPONSE</td>
<td></td>
</tr>
<tr>
<td>0x1c - 0x1d</td>
<td>CONNECTION_CLOSE</td>
<td></td>
</tr>
</tbody>
</table>
Frames

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Frame Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>PADDING</td>
</tr>
<tr>
<td>0x01</td>
<td>PING</td>
</tr>
<tr>
<td>0x02 - 0x03</td>
<td>ACK</td>
</tr>
<tr>
<td>0x04</td>
<td>RESET_STREAM</td>
</tr>
<tr>
<td>0x05</td>
<td>STOP_SENDING</td>
</tr>
<tr>
<td>0x06</td>
<td>CRYPTO</td>
</tr>
<tr>
<td>0x07</td>
<td>NEW_TOKEN</td>
</tr>
<tr>
<td>0x08 - 0x0f</td>
<td>STREAM</td>
</tr>
<tr>
<td>0x10</td>
<td>MAX_DATA</td>
</tr>
<tr>
<td>0x11</td>
<td>MAX_STREAM_DATA</td>
</tr>
<tr>
<td>0x12 - 0x13</td>
<td>MAX_STREAMS</td>
</tr>
<tr>
<td>0x14</td>
<td>DATA_BLOCKED</td>
</tr>
<tr>
<td>0x15</td>
<td>STREAM_DATA_BLOCKED</td>
</tr>
<tr>
<td>0x16 - 0x17</td>
<td>STREAMS_BLOCKED</td>
</tr>
<tr>
<td>0x18</td>
<td>NEW_CONNECTION_ID</td>
</tr>
<tr>
<td>0x19</td>
<td>RETIRE_CONNECTION_ID</td>
</tr>
<tr>
<td>0x1a</td>
<td>PATH_CHALLENGE</td>
</tr>
<tr>
<td>0x1b</td>
<td>PATH_RESPONSE</td>
</tr>
<tr>
<td>0x1c - 0x1d</td>
<td>CONNECTION_CLOSE</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+---</td>
<td>---</td>
</tr>
<tr>
<td>Stream ID (i)</td>
<td>...</td>
</tr>
<tr>
<td>+---</td>
<td>---</td>
</tr>
<tr>
<td>[Offset (i)]</td>
<td>...</td>
</tr>
<tr>
<td>+---</td>
<td>---</td>
</tr>
<tr>
<td>[Length (i)]</td>
<td>...</td>
</tr>
<tr>
<td>+---</td>
<td>---</td>
</tr>
<tr>
<td>Stream Data (*)</td>
<td>...</td>
</tr>
</tbody>
</table>
QUIC Packetization: Example

QUIC Packet

- Header = 0b01
- Spin Bit
- Dest Conn ID
- Key Phase
- Packet Number
QUIC Packetization: Example

- **Header = 0b01**
- **Spin Bit**
- **Dest Conn ID**
- **Key Phase**
- **Packet Number**

QUIC Packet

- STREAM Frame
- STREAM Frame
- ACK Frame
QUIC Packetization: Example

- **Header = 0b01**
- **Spin Bit**
- **Dest Conn ID**
- **Key Phase**
- **PN = 56**

** STREAM Frame **
- Stream ID: 5
- Offset: 1123
- Length: 500
- Fin: False

** Application Data **

** ACK Frame **

** QUIC Packet **
QUIC Packetization: Example

QUIC Packet

- Header = 0b01
- Spin Bit
- Dest Conn ID
- Key Phase
- PN = 56

STREAM Frame
- Stream ID: 5
- Offset: 1123
- Length: 500
- Fin: False
  Application Data

STREAM Frame
- Stream ID: 8
- Length: 300
- Fin: False
  Application Data

ACK Frame
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Largest Acknowledged</td>
<td>(i)</td>
</tr>
<tr>
<td>ACK Delay</td>
<td>(i)</td>
</tr>
<tr>
<td>ACK Range Count</td>
<td>(i)</td>
</tr>
<tr>
<td>First ACK Range</td>
<td>(i)</td>
</tr>
<tr>
<td>ACK Ranges</td>
<td>(*)</td>
</tr>
<tr>
<td>[ECN Counts]</td>
<td></td>
</tr>
</tbody>
</table>
ACK Frame

Highest Packet Number seen so far
ACK Frame

Highest Packet Number seen so far

Time since Largest Acked was received
### ACK 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>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Highest Packet Number seen so far**
- **Time since Largest Acked was received**
- **Contiguous from Largest Acked**

<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>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Largest Acknowledged (i)**
- **ACK Delay (i)**
- **ACK Range Count (i)**
- **First ACK Range (i)**
- **ACK Ranges (*)**
- **[ECN Counts]**
QUIC Packetization: Ack Example

Packets received: 1 ... 125
Time since largest received: 25ms
  represented as a shifted value (default 3, negotiable)
  25ms = 3125us << 3

ACK fields
Largest packet received so far: 125
First Ack Range: 124
Ack Range Count: 0
QUIC Packetization: Example

QUIC Packet:
- Header = 0b01
- Spin Bit
- Dest Conn ID
- Key Phase
- PN = 56

STREAM Frame:
- Stream ID: 5
- Offset: 1123
- Length: 500
- Fin: False
  - Application Data

STREAM Frame:
- Stream ID: 8
- Length: 300
- Fin: False
  - Application Data

ACK Frame:
- Largest: 125
- Ack Delay: 3125
- Ack Range: 0
- First Range: 124
QUIC Packetization: Ack Example

Packets received: 1 … 125, 130
Time since largest received: 0ms

ACK fields
Largest packet received so far: 130
First Ack Range: 0
Gap #1: 126 - 129
Ack Range #1: 1 - 125
QUIC Packetization: Ack Example

Packets received: 1 ... 125, 130
Time since largest received: 0ms

ACK fields
Largest packet received so far: 130
First Ack Range: 0
Gap #1: 126 - 129 (encoded as 3)
Ack Range #1: 1 - 125 (encoded as 124)
QUIC Packetization: Example

QUIC Packet

- **Header = 0b01**
- **Spin Bit**
- **Dest Conn ID**
- **Key Phase**
- **PN = 56**

STREAM Frame

- **Stream ID:** 5
- **Offset:** 1123
- **Length:** 500

Application Data

STREAM Frame

- **Stream ID:** 8
- **Length:** 300

Application Data

ACK Frame

- **Largest:** 130
- **Ack Delay:** 0
- **Range Count:** 1
- **First Range:** 0
- **Gap #1:** 3
- **Range #1:** 124
QUIC Packetization: Loss Example

Packet 56 dropped
QUIC Packetization: Loss Example

Packet 56 dropped

Also, Stream 8 was reset
Packet 56 dropped

Also, Stream 8 was reset

QUIC loss detection marks packet 56 as lost
  let’s say last packet sent was packet number 74
QUIC Packetization: Example

QUIC Packet

- **Header = 0b01**
- **Spin Bit**
- **Dest Conn ID**
- **Key Phase**
- **PN = 56**

**STREAM Frame**
- Stream ID: 5
- Offset: 1123
- Length: 500

**STREAM Frame**
- Stream ID: 8
- Length: 300

**ACK Frame**
- Largest: 130
- Ack Delay: 0
- Range Count: 1
- First Range: 0
- Gap #1: 3
- Range #1: 124

**Application Data**

**Application Data**
QUIC Packetization: Example

QUIC Packet

- Header = 0b01
- Spin Bit
- Dest Conn ID
- Key Phase
- PN = 75

STREAM Frame
- Stream ID: 5
- Offset: 1123
- Length: 500

Application Data

STREAM Frame
- Stream ID: 8
- Length: 300

Application Data

ACK Frame
- Largest: 130
- Ack Delay: 0
- Range Count: 1
- First Range: 0
- Gap #1: 3
- Range #1: 124
**QUIC Packetization: Example**

- **Header = 0b01**
- **Spin Bit**
- **Dest Conn ID**
- **Key Phase**
- **PN = 75**

**QUIC Packet**

- **STREAM Frame**
  - Stream ID: 5
  - Offset: 1123
  - Length: 500
  - Application Data
- **STREAM Frame**
  - Stream ID: 8
  - Length: 300
  - Application Data
- **ACK Frame**
  - Largest: 130
  - Ack Delay: 0
  - Range Count: 1
  - First Range: 0
  - Gap #1: 7
  - Range #1: 124
QUIC Loss Recovery

Packet numbers represent transmission order stream IDs and offsets used for delivery order monotonically increasing 62-bit packet numbers (caveat: multiple PN spaces during connection setup)
QUIC Loss Recovery

Packet numbers represent transmission order
stream IDs and offsets used for delivery order
monotonically increasing 62-bit packet numbers
(caveat: multiple PN spaces during connection setup)

Packets are containers
carry a mix of various types of frames
QUIC Loss Recovery

Packet numbers represent transmission order
stream IDs and offsets used for delivery order
monotonically increasing 62-bit packet numbers
(caveat: multiple PN spaces during connection setup)

Packets are containers
carry a mix of various types of frames

Retransmissions are not automatically high priority
depends on relative stream priority
application-dependent
Generating ACK frames

SHOULD ACK every other packet
subject to 25ms delayed ack timer
Generating ACK frames

SHOULD ACK every other packet
subject to 25ms delayed ack timer

SHOULD ACK immediately if:
Received packet number != largest received + 1
CE codepoint received
Generating ACK frames

SHOULD ACK every other packet
subject to 25ms delayed ack timer

SHOULD ACK immediately if:
Received packet number != largest received + 1
CE codepoint received

MAY process more packets before ACK
allows less frequent acking
QUIC Loss Detection

Loss detection only when ACK frame received
that newly acks a packet
use both packet and time thresholds
QUIC Loss Detection

Loss detection only when ACK frame received that newly acks a packet
use both packet and time thresholds

Packet threshold
reordering $\geq$ 3 packets
QUIC Loss Detection

Loss detection only when ACK frame received that newly acks a packet
use both packet and time thresholds

Packet threshold
reordering \( \geq 3 \) packets

Time threshold
reordering \( \geq 1 \) packet AND
time \( > \frac{9}{8} \times \max(\text{SRTT}, \text{latest}_\text{RTT}) \)
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data) restarted when new ACK-eliciting packet (tail) sent exponential backoff (pto *= 2)

\[ pto = \text{smoothed}_\text{rtt} + \max(4*\text{rttvar}, k\text{Granularity}) + \max_\text{ack}_\text{delay} \]

Timeout does not necessarily mean packet loss exception: if no data to send, mark outstanding as lost
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data)
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data) restarted when new ACK-eliciting packet (tail) sent
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data) restarted when new ACK-eliciting packet (tail) sent exponential backoff (pto *= 2)
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data) restarted when new ACK-eliciting packet (tail) sent exponential backoff (pto *= 2)

pto = smoothed_rtt + max(4*rttvar, kGranularity) + max_ack_delay
No ACKs received: Probe Timeout

Probe Timeout (PTO) triggers packet(s) when no ACK on PTO, send 1 or 2 probe packets (new or old data) restarted when new ACK-eliciting packet (tail) sent exponential backoff (pto *= 2)

\[
\text{pto} = \text{smoothed}_\text{rtt} + \max(4*\text{rttvar}, \text{kGranularity}) + \max_\text{ack}_\text{delay}
\]

Timeout does not necessarily mean packet loss exception: if no data to send, mark outstanding as lost
No ACKs received: Persistent Congestion

When all packets are lost over a long-enough time
(smoothed_rtt + 4 * rttvar + max_ack_delay) *
(2 ^ kPersistentCongestionThreshold - 1)
No ACKs received: Persistent Congestion

When all packets are lost over a long-enough time
(smoothed_rtt + 4 * rttvar + max_ack_delay) *
(2 ^ kPersistentCongestionThreshold - 1)

Collapse congestion window to min_cwnd
No ACKs received: Persistent Congestion

When all packets are lost over a long-enough time
(smoothed_rtt + 4 * rttvar + max_ack_delay) *
(2 ^ kPersistentCongestionThreshold - 1)

Collapse congestion window to min_cwnd

Default kPersistentCongestionThreshold = 2
same as 2 TLPs followed by an RTO
Tooling

In-network packet tracing
Wireshark dissector available
This isn’t enough. Why?

Endpoint-based packet tracing
Log packet and frame details at endpoint
(also log other transport info, such as cwnd)
quic-trace
QUICvis
Tooling: quic-trace

Written by Victor Vasiliev et al (Google)
Available at https://github.com/google/quic-trace
Input: protobuf or JSON
Tooling: QUICvis

Written by Robin Marx et al
Available at https://quic.edm.uhasselt.be/
Input: JSON