

dotnet / aspnetcore Public

- <> Code
- Issues 3.6k
- Pull requests 276
- Discussions
- Actions
- Projects

# Commit 96ccc40



BrennanConroy committed on Oct 8, 2025

Merged PR 54041: Fix Http3 Encoder/Decoder stream exit condition



[darc-release/9.0-0725b487-be56-4f09-9589-1f33106ab4c9](#) (#66371, #64312) +
   
[release/9.0](#) (#64312)

v9.0.15 ...
   
 v9.0.11

1 parent [0fbde91](#) commit 96ccc40

3 files changed

+77 -21

Top

Filter files...

- src/Servers/Kestrel
  - Core/src/Internal/Http3
    - Http3ControlStream.cs**
  - shared/test/Http3
    - Http3InMemory.cs
  - test/InMemory.FunctionalTests/Http3
    - Http3ConnectionTests.cs

Search within code

...re/src/Internal/Http3/Http3ControlStream.cs



```

@@ -65,6 +65,12 @@ public Http3ControlStream(Http3StreamContext context,
long? headerType)

```

```

65      65      context.ClientPeerSettings,

```

```

66 66         this);
67 67         _frameWriter.Reset(context.Transport.Output, context.ConnectionId);
68 +
69 +         _streamClosedFeature.OnClosed(static state =>
70 +         {
71 +             var stream = (Http3ControlStream)state!;
72 +             stream.OnStreamClosed();
73 +         }, this);
68 74     }
69 75
70 76     private void OnStreamClosed()
77
78     @@ -135,12 +141,6 @@ private bool TryClose()
79
135 141
136 142     internal async ValueTask ProcessOutboundSendsAsync(long id)
137 143     {
138 -         _streamClosedFeature.OnClosed(static state =>
139 -         {
140 -             var stream = (Http3ControlStream)state!;
141 -             stream.OnStreamClosed();
142 -         }, this);
143 -
144 144         await _frameWriter.WriteStreamIdAsync(id);
145 145         await
146         _frameWriter.WriteSettingsAsync(_serverPeerSettings.GetNonProtocolDefaults());
146 146     }
147
148     @@ -311,18 +311,13 @@ private async Task HandleControlStream()
149
311 311     }
312 312     }
313 313
314 -     private async ValueTask HandleEncodingDecodingTask()
314 +     private Task HandleEncodingDecodingTask()
315 315     {
316 316         // Noop encoding and decoding task. Settings make it so we don't need
317         // to read content of encoder and decoder.
318 317         // An endpoint MUST allow its peer to create an encoder stream and a
319 318         // decoder stream even if the connection's settings prevent their use.
319 319
320 -     while (_isClosed == 0)

```

```

321     -         {
322     -             var result = await Input.ReadAsync();
323     -             var readableBuffer = result.Buffer;
324     -             Input.AdvanceTo(readableBuffer.End);
325     -         }
320     +         return Input.CopyToAsync(Stream.Null);
326     321     }
327     322
328     323     private ValueTask ProcessHttp3ControlStream(Http3RawFrame incomingFrame,
bool isContinuedFrame, in ReadOnlySequence<byte> payload, out SequencePosition
consumed)
@@ -372,11 +367,6 @@ private ValueTask ProcessSettingsFrameAsync(bool
isContinuedFrame, ReadOnlySeque
372     367     }
373     368
374     369     _haveReceivedSettingsFrame = true;
375     -     _streamClosedFeature.OnClosed(static state =>
376     -     {
377     -         var stream = (Http3ControlStream)state!;
378     -         stream.OnStreamClosed();
379     -     }, this);
380     370     }
381     371
382     372     while (true)

```

```

.../Kestrel/shared/test/Http3/Http3InMemory.cs
@@ -301,12 +301,22 @@ public void
OnInboundControlStreamSetting(Http3SettingType type, long value)
301     301
302     302     public bool
OnInboundDecoderStream(Server.Kestrel.Core.Internal.Http3.Http3ControlStream
stream)
303     303     {
304     -     return _inner.OnInboundDecoderStream(stream);
304     +     var res = _inner.OnInboundDecoderStream(stream);
305     +     if (_http3TestBase._runningStreams.TryGetValue(stream.StreamId, out
var testStream))
306     +     {
307     +         testStream.OnDecoderStreamCreatedTcs.TrySetResult();

```

308	+	}
309	+	return res;
305	310	}
306	311	
307	312	public bool OnInboundEncoderStream(Server.Kestrel.Core.Internal.Http3.Http3ControlStream stream)
308	313	{
309	-	return _inner.OnInboundEncoderStream(stream);
314	+	var res = _inner.OnInboundEncoderStream(stream);
315	+	if (_http3TestBase._runningStreams.TryGetValue(stream.StreamId, out var testStream))
316	+	{
317	+	testStream.OnEncoderStreamCreatedTcs.TrySetResult();
318	+	}
319	+	return res;
310	320	}
311	321	
312	322	public void OnStreamCompleted(IHttp3Stream stream)
⌵ ⌶		@@ -480,6 +490,8 @@ internal class Http3StreamBase
480	490	internal TaskCompletionSource OnStreamCreatedTcs { get; } = new TaskCompletionSource(TaskCreationOptions.RunContinuationsAsynchronously);
481	491	internal TaskCompletionSource OnStreamCompletedTcs { get; } = new TaskCompletionSource(TaskCreationOptions.RunContinuationsAsynchronously);
482	492	internal TaskCompletionSource OnHeaderReceivedTcs { get; } = new TaskCompletionSource(TaskCreationOptions.RunContinuationsAsynchronously);
493	+	internal TaskCompletionSource OnDecoderStreamCreatedTcs { get; } = new TaskCompletionSource(TaskCreationOptions.RunContinuationsAsynchronously);
494	+	internal TaskCompletionSource OnEncoderStreamCreatedTcs { get; } = new TaskCompletionSource(TaskCreationOptions.RunContinuationsAsynchronously);
483	495	
484	496	internal TestStreamContext StreamContext { get; }
485	497	internal DuplexPipe.DuplexPipePair Pair { get; }
⌵ ⌶		@@ -496,6 +508,8 @@ public long Error
496	508	public Task OnStreamCreatedTask => OnStreamCreatedTcs.Task;
497	509	public Task OnStreamCompletedTask => OnStreamCompletedTcs.Task;
498	510	public Task OnHeaderReceivedTask => OnHeaderReceivedTcs.Task;
511	+	public Task OnDecoderStreamCreatedTask => OnDecoderStreamCreatedTcs.Task;
512	+	public Task OnEncoderStreamCreatedTask => OnEncoderStreamCreatedTcs.Task;

```

499 513
500 514     public ConnectionAbortedException AbortReadException =>
        StreamContext.AbortReadException;
501 515     public ConnectionAbortedException AbortWriteException =>
        StreamContext.AbortWriteException;

```



...nctionalTests/Http3/Http3ConnectionTests.cs



@@ -18,7 +18,6 @@

```

18 18     using Microsoft.Extensions.Primitives;
19 19     using Microsoft.Net.Http.Headers;
20 20     using Http3SettingType =
        Microsoft.AspNetCore.Server.Kestrel.Core.Internal.Http3.Http3SettingType;
21 - using Microsoft.AspNetCore.Server.Kestrel.Core.Internal.Infrastructure;
22 21
23 22     namespace Microsoft.AspNetCore.Server.Kestrel.Core.Tests;
24 23

```



@@ -768,6 +767,59 @@ public async Task ErrorCodeIsValidOnConnectionTimeout()

```

768 767         Assert.InRange(errorCodeFeature.Error, 0, (1L << 62) - 1); // Valid
        range for HTTP/3 error codes
769 768     }
770 769
770 +     [Theory]
771 +     [InlineData(2)] // encoder
772 +     [InlineData(3)] // decoder
773 +     public async Task IgnoredControlStreams_CloseConnectionOnEndStream(int
        streamType)
774 +     {
775 +         await Http3Api.InitializeConnectionAsync(_noopApplication);
776 +
777 +         var stream = await Http3Api.CreateControlStream(streamType);
778 +
779 +         // PipeWriter will be completed when end of stream is received. Should
        exit read loop and close stream
780 +         // which will cause the connection to close with an error.
781 +         await stream.SendFrameAsync(Http3FrameType.Data, Memory<byte>.Empty,
        endStream: true);
782 +
783 +         await stream.OnStreamCompletedTask.DefaultTimeout();

```

```
784 +
785 +     Http3Api.TriggerTick();
786 +     Http3Api.TriggerTick(TimeSpan.FromSeconds(1));
787 +
788 +     await
    Http3Api.WaitForConnectionErrorAsync<Http3ConnectionErrorException>(
789 +         ignoreNonGoAwayFrames: true,
790 +         expectedLastStreamId: 0,
791 +         expectedErrorCode: Http3ErrorCode.ClosedCriticalStream,
792 +         matchExpectedErrorMessage: AssertExpectedErrorMessages,
793 +         expectedErrorMessage: CoreStrings.Http3ErrorControlStreamClosed);
794 +         MetricsAssert.Equal(ConnectionEndReason.ClosedCriticalStream,
    Http3Api.ConnectionTags);
795 +     }
796 +
797 +     [Theory]
798 +     [InlineData(2)] // encoder
799 +     [InlineData(3)] // decoder
800 +     public async Task IgnoredControlStreams_CloseConnectionOnStreamClose(int
    streamType)
801 +     {
802 +         await Http3Api.InitializeConnectionAsync(_noopApplication);
803 +
804 +         var stream = await Http3Api.CreateControlStream(streamType);
805 +
806 +         await (streamType == 2 ? stream.OnEncoderStreamCreatedTask :
    stream.OnDecoderStreamCreatedTask).DefaultTimeout();
807 +
808 +         // Simulate quic layer closing the stream
809 +         stream.StreamContext.Close();
810 +
811 +         Http3Api.TriggerTick();
812 +         Http3Api.TriggerTick(TimeSpan.FromSeconds(1));
813 +
814 +         await
    Http3Api.WaitForConnectionErrorAsync<Http3ConnectionErrorException>(
815 +             ignoreNonGoAwayFrames: true,
816 +             expectedLastStreamId: 0,
817 +             expectedErrorCode: Http3ErrorCode.ClosedCriticalStream,
818 +             matchExpectedErrorMessage: AssertExpectedErrorMessages,
```

```
819 +         expectedErrorMessage: CoreStrings.Http3ErrorControlStreamClosed);
820 +         MetricsAssert.Equal(ConnectionEndReason.ClosedCriticalStream,
      Http3Api.ConnectionTags);
821 +     }
822 +
771 823     private sealed class ThrowingMultiplexedConnectionContext :
      TestMultiplexedConnectionContext
772 824     {
773 825         private int _skipCount;
```



## Comments 0



Please [sign in](#) to comment.