

When Mask Is Present (When Mask == 1)

Payload Length < 126 Bytes

Note The payload is all the bytes that come after the masking key. DO NOT assume that it is exactly 4 bytes after the masking key

<i>FIN</i>	<i>RSV1</i>	<i>RSV2</i>	<i>RSV3</i>	<i>Opcode</i>			
1	0	0	0	0	0	0	0

Masking Key							
1	0	1	0	1	1	0	0

Masking Key Cont.							
0	0	1	0	0	1	0	0

Payload							
0	1	1	0	1	0	1	0

<i>Payload Continued</i>
... Rest of the WebSocket Frame ...

Mask	Payload Length						
1	0	0	1	0	1	1	0

Masking Key Cont.							
0	1	1	0	0	1	0	0

Masking Key Cont.							
0	0	1	0	0	1	0	0

<i>Payload Continued</i>							
0	1	0	0	0	1	1	0

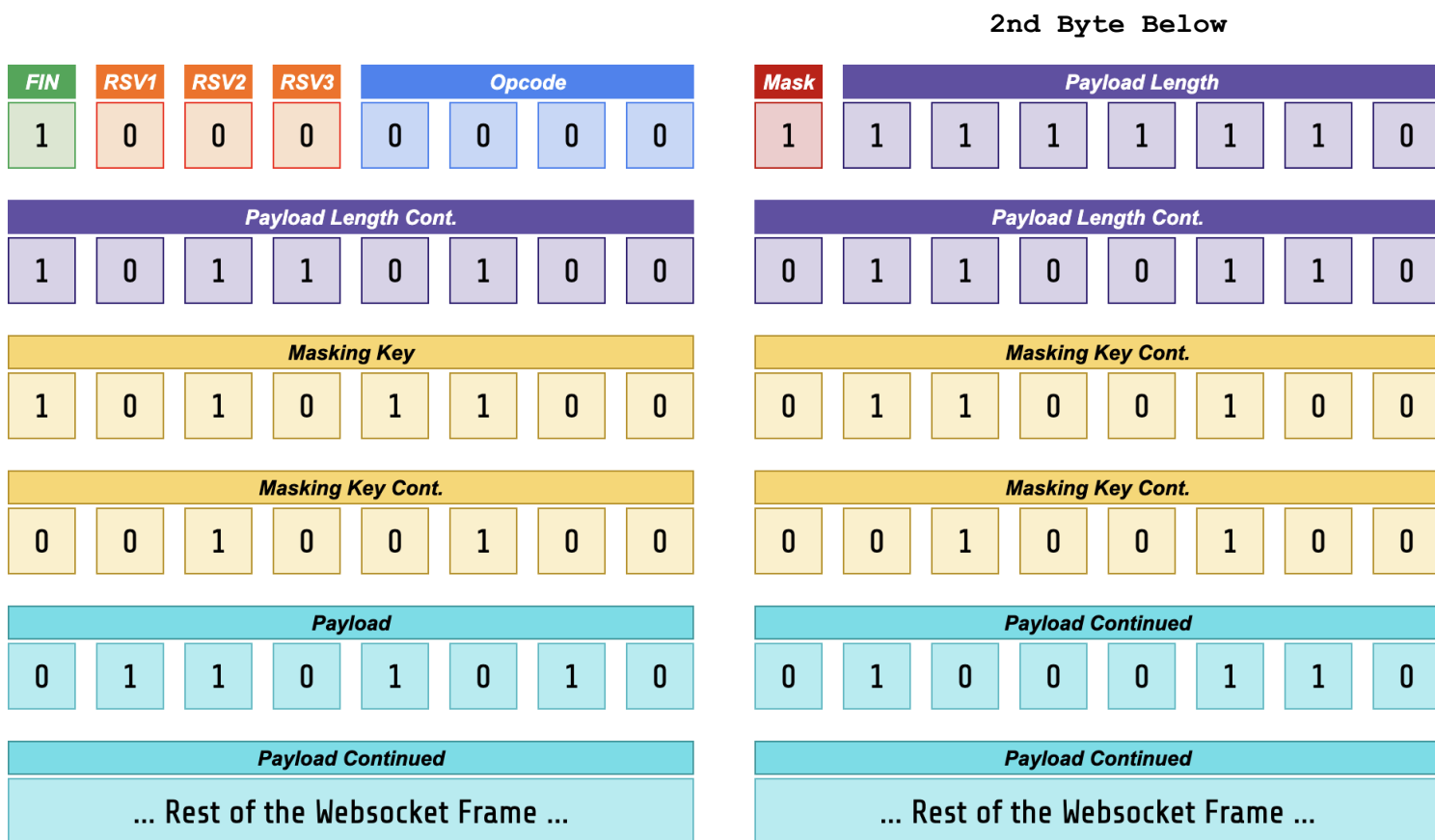
<i>Payload Continued</i>
... Rest of the WebSocket Frame ...

Payload Length >= 126 and < 65536 Bytes

Note 1 The payload length in the 2nd byte will be exactly 126 (1111110)

Note 2 The next 16 bits (2 bytes) after the 2nd byte is the actual payload length

Note 3 The payload is all the bytes that come after the masking key. DO NOT assume that it is exactly 4 bytes after the masking key



Payload Length >= 65536 Bytes

Note 1 The payload length in the 2nd byte will be exactly 127 (1111111)

Note 2 The next 64 bits (8 bytes) after the 2nd byte is the actual payload length

Note 3 The payload is all the bytes that come after the masking key. DO NOT assume that it is exactly 4 bytes after the masking key

2nd Byte Below

FIN	RSV1	RSV2	RSV3	Opcode			
1	0	0	0	0	0	0	0

Payload Length Cont.							
1	0	1	1	0	1	0	0

Payload Length Cont.							
0	0	1	0	1	1	0	0

Payload Length Cont.							
0	1	1	0	1	0	0	0

Payload Length Cont.							
0	0	0	1	1	0	0	0

Masking Key							
1	0	1	0	1	1	0	0

Masking Key Cont.							
0	0	1	0	0	1	0	0

Payload							
0	1	1	0	1	0	1	0

<i>Payload Continued</i>
... Rest of the WebSocket Frame ...

Mask	Payload Length						
1	1	1	1	1	1	1	1

Payload Length Cont.							
0	1	1	0	0	1	1	0

Payload Length Cont.							
0	1	0	1	0	1	1	0

Payload Length Cont.							
0	1	1	1	0	0	0	0

Payload Length Cont.							
0	0	1	0	0	1	1	0

Masking Key Cont.							
0	1	1	0	0	1	0	0

Masking Key Cont.							
0	0	1	0	0	1	0	0

Payload Continued							
0	1	0	0	0	1	1	0

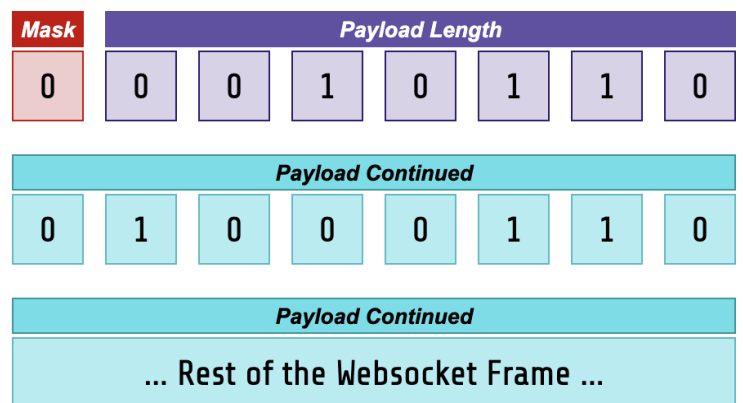
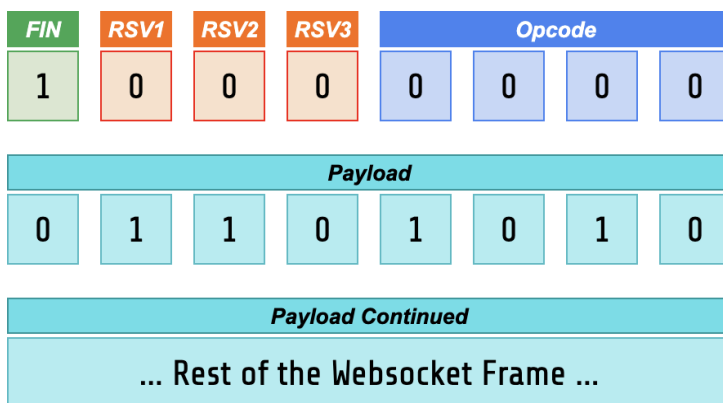
<i>Payload Continued</i>
... Rest of the WebSocket Frame ...

When Mask Is Not Present (When Mask == 0)

Payload Length < 126 Bytes

Note 1 The payload is all the bytes that come after the payload length. DO NOT assume that it is exactly 4 bytes after the payload length

Note 2 There is no masking key since mask is 0



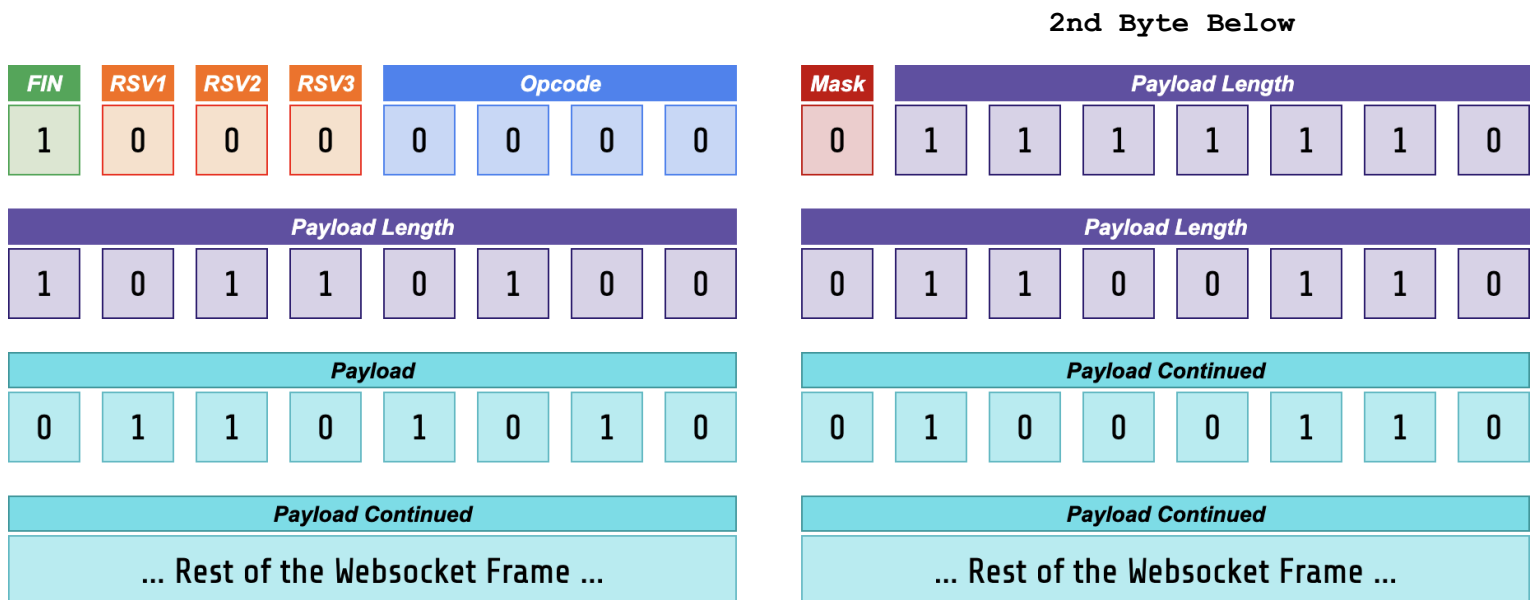
Payload Length ≥ 126 and < 65536 Bytes

Note 1 The payload length in the 2nd byte will be exactly 126 (1111110)

Note 2 The next 16 bits (2 bytes) after the 2nd byte is the actual payload length

Note 3 The payload is all the bytes that come after the payload length. DO NOT assume that it is exactly 4 bytes after the payload length

Note 4 There is no masking key since mask is 0



Payload Length >= 65536 Bytes

Note 1 The payload length in the 2nd byte will be exactly 127 (1111111)

Note 2 The next 64 bits (8 bytes) after the 2nd byte is the actual payload length

Note 3 The payload is all the bytes that come after the payload length. DO NOT assume that it is exactly 4 bytes after the payload length

Note 4 There is no masking key since mask is 0

2nd Byte Below

FIN	RSV1	RSV2	RSV3	Opcode			
1	0	0	0	0	0	0	0

Payload Length Cont.							
1	0	1	1	0	1	0	0

Payload Length Cont.						
0	0	1	0	1	1	0

Payload Length Cont.							
0	1	1	0	1	0	0	0

Payload Length Cont.						
0	0	0	1	1	0	0

Payload							
0	1	1	0	1	0	1	0

<i>Payload Continued</i>
... Rest of the WebSocket Frame ...

Mask	Payload Length						
0	1	1	1	1	1	1	1

Payload Length Cont.							
0	1	1	0	0	1	1	0

Payload Length Cont.							
0	1	0	1	0	1	1	0

Payload Length Cont.							
0	1	1	1	0	0	0	0

Payload Length Cont.							
0	0	1	0	0	1	1	0

Payload Continued							
0	1	0	0	0	1	1	0

The diagram shows a rectangular frame divided into two horizontal sections. The top section is light blue and contains the text *Payload Continued*. The bottom section is white and contains the text *... Rest of the WebSocket Frame ...*.

Good
Luck

