

# WebSocket Implementation Test Report



Summary report generated on 2014-07-16T21:49:11Z (UTC) by [Autobahn WebSockets Testsuite](#) v0.6.1/v0.8.8.

Pass	Test case was executed and passed successfully.
Non-Strict	Test case was executed and passed non-strictly. A non-strict behavior is one that does not adhere to a SHOULD-behavior as described in the protocol specification or a well-defined, canonical behavior that appears to be desirable but left open in the protocol specification. An implementation with non-strict behavior is still conformant to the protocol specification.
Fail	Test case was executed and failed. An implementation which fails a test case - other than a performance/limits related one - is non-conforming to a MUST-behavior as described in the protocol specification.
Info	Informational test case which detects certain implementation behavior left unspecified by the spec but nevertheless potentially interesting to implementors.
Missing	Test case is missing, either because it was skipped via the test suite configuration or deactivated, i.e. because the implementation does not implement the tested feature or breaks during running the test case.

1 Framing	dmc_websockets/1.2.0	
1.1 Text Messages		
<a href="#">Case 1.1.1</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.2</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.3</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.4</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.5</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.6</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.7</a>	<a href="#">Pass</a>	1000
<a href="#">Case 1.1.8</a>	<a href="#">Pass</a>	1000
1 Framing	dmc_websockets/1.2.0	
1.2 Binary Messages		
<a href="#">Case 1.2.1</a>	Missing	
<a href="#">Case 1.2.2</a>	Missing	
<a href="#">Case 1.2.3</a>	Missing	
<a href="#">Case 1.2.4</a>	Missing	
<a href="#">Case 1.2.5</a>	Missing	
<a href="#">Case 1.2.6</a>	Missing	
<a href="#">Case 1.2.7</a>	Missing	
<a href="#">Case 1.2.8</a>	Missing	
2 Pings/Pongs	dmc_websockets/1.2.0	
<a href="#">Case 2.1</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.2</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.3</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.4</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.5</a>	<a href="#">Pass</a>	1002
<a href="#">Case 2.6</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.7</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.8</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.9</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.10</a>	<a href="#">Pass</a>	1000
<a href="#">Case 2.11</a>	<a href="#">Pass</a>	1000

3 Reserved Bits	dmc_websockets/1.2.0	
<a href="#">Case 3.1</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.2</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.3</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.4</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.5</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.6</a>	<a href="#">Pass</a>	1002
<a href="#">Case 3.7</a>	<a href="#">Pass</a>	1002
4 Opcodes	dmc_websockets/1.2.0	
4.1 Non-control Opcodes		
<a href="#">Case 4.1.1</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.1.2</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.1.3</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.1.4</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.1.5</a>	<a href="#">Pass</a>	1002
4 Opcodes	dmc_websockets/1.2.0	
4.2 Control Opcodes		
<a href="#">Case 4.2.1</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.2.2</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.2.3</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.2.4</a>	<a href="#">Pass</a>	1002
<a href="#">Case 4.2.5</a>	<a href="#">Pass</a>	1002
5 Fragmentation	dmc_websockets/1.2.0	
<a href="#">Case 5.1</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.2</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.3</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.4</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.5</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.6</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.7</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.8</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.9</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.10</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.11</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.12</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.13</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.14</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.15</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.16</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.17</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.18</a>	<a href="#">Pass</a>	1002
<a href="#">Case 5.19</a>	<a href="#">Pass</a>	1000
<a href="#">Case 5.20</a>	<a href="#">Pass</a>	1000
6 UTF-8 Handling	dmc_websockets/1.2.0	
6.1 Valid UTF-8 with zero payload fragments		
<a href="#">Case 6.1.1</a>	Missing	
<a href="#">Case 6.1.2</a>	Missing	
<a href="#">Case 6.1.3</a>	Missing	

6 UTF-8 Handling	dmc_websockets/1.2.0
6.2 Valid UTF-8 unfragmented, fragmented on code-points and within code-points	
<a href="#">Case 6.2.1</a>	Missing
<a href="#">Case 6.2.2</a>	Missing
<a href="#">Case 6.2.3</a>	Missing
<a href="#">Case 6.2.4</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.3 Invalid UTF-8 differently fragmented	
<a href="#">Case 6.3.1</a>	Missing
<a href="#">Case 6.3.2</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.4 Fail-fast on invalid UTF-8	
<a href="#">Case 6.4.1</a>	Missing
<a href="#">Case 6.4.2</a>	Missing
<a href="#">Case 6.4.3</a>	Missing
<a href="#">Case 6.4.4</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.5 Some valid UTF-8 sequences	
<a href="#">Case 6.5.1</a>	Missing
<a href="#">Case 6.5.2</a>	Missing
<a href="#">Case 6.5.3</a>	Missing
<a href="#">Case 6.5.4</a>	Missing
<a href="#">Case 6.5.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.6 All prefixes of a valid UTF-8 string that contains multi-byte code points	
<a href="#">Case 6.6.1</a>	Missing
<a href="#">Case 6.6.2</a>	Missing
<a href="#">Case 6.6.3</a>	Missing
<a href="#">Case 6.6.4</a>	Missing
<a href="#">Case 6.6.5</a>	Missing
<a href="#">Case 6.6.6</a>	Missing
<a href="#">Case 6.6.7</a>	Missing
<a href="#">Case 6.6.8</a>	Missing
<a href="#">Case 6.6.9</a>	Missing
<a href="#">Case 6.6.10</a>	Missing
<a href="#">Case 6.6.11</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.7 First possible sequence of a certain length	
<a href="#">Case 6.7.1</a>	Missing
<a href="#">Case 6.7.2</a>	Missing
<a href="#">Case 6.7.3</a>	Missing
<a href="#">Case 6.7.4</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.8 First possible sequence length 5/6 (invalid codepoints)	
<a href="#">Case 6.8.1</a>	Missing
<a href="#">Case 6.8.2</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0

6.9 Last possible sequence of a certain length	
<a href="#">Case 6.9.1</a>	Missing
<a href="#">Case 6.9.2</a>	Missing
<a href="#">Case 6.9.3</a>	Missing
<a href="#">Case 6.9.4</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.10 Last possible sequence length 4/5/6 (invalid codepoints)	
<a href="#">Case 6.10.1</a>	Missing
<a href="#">Case 6.10.2</a>	Missing
<a href="#">Case 6.10.3</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.11 Other boundary conditions	
<a href="#">Case 6.11.1</a>	Missing
<a href="#">Case 6.11.2</a>	Missing
<a href="#">Case 6.11.3</a>	Missing
<a href="#">Case 6.11.4</a>	Missing
<a href="#">Case 6.11.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.12 Unexpected continuation bytes	
<a href="#">Case 6.12.1</a>	Missing
<a href="#">Case 6.12.2</a>	Missing
<a href="#">Case 6.12.3</a>	Missing
<a href="#">Case 6.12.4</a>	Missing
<a href="#">Case 6.12.5</a>	Missing
<a href="#">Case 6.12.6</a>	Missing
<a href="#">Case 6.12.7</a>	Missing
<a href="#">Case 6.12.8</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.13 Lonely start characters	
<a href="#">Case 6.13.1</a>	Missing
<a href="#">Case 6.13.2</a>	Missing
<a href="#">Case 6.13.3</a>	Missing
<a href="#">Case 6.13.4</a>	Missing
<a href="#">Case 6.13.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.14 Sequences with last continuation byte missing	
<a href="#">Case 6.14.1</a>	Missing
<a href="#">Case 6.14.2</a>	Missing
<a href="#">Case 6.14.3</a>	Missing
<a href="#">Case 6.14.4</a>	Missing
<a href="#">Case 6.14.5</a>	Missing
<a href="#">Case 6.14.6</a>	Missing
<a href="#">Case 6.14.7</a>	Missing
<a href="#">Case 6.14.8</a>	Missing
<a href="#">Case 6.14.9</a>	Missing
<a href="#">Case 6.14.10</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.15 Concatenation of incomplete sequences	

<a href="#">Case 6.15.1</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.16 Impossible bytes	
<a href="#">Case 6.16.1</a>	Missing
<a href="#">Case 6.16.2</a>	Missing
<a href="#">Case 6.16.3</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.17 Examples of an overlong ASCII character	
<a href="#">Case 6.17.1</a>	Missing
<a href="#">Case 6.17.2</a>	Missing
<a href="#">Case 6.17.3</a>	Missing
<a href="#">Case 6.17.4</a>	Missing
<a href="#">Case 6.17.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.18 Maximum overlong sequences	
<a href="#">Case 6.18.1</a>	Missing
<a href="#">Case 6.18.2</a>	Missing
<a href="#">Case 6.18.3</a>	Missing
<a href="#">Case 6.18.4</a>	Missing
<a href="#">Case 6.18.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.19 Overlong representation of the NUL character	
<a href="#">Case 6.19.1</a>	Missing
<a href="#">Case 6.19.2</a>	Missing
<a href="#">Case 6.19.3</a>	Missing
<a href="#">Case 6.19.4</a>	Missing
<a href="#">Case 6.19.5</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.20 Single UTF-16 surrogates	
<a href="#">Case 6.20.1</a>	Missing
<a href="#">Case 6.20.2</a>	Missing
<a href="#">Case 6.20.3</a>	Missing
<a href="#">Case 6.20.4</a>	Missing
<a href="#">Case 6.20.5</a>	Missing
<a href="#">Case 6.20.6</a>	Missing
<a href="#">Case 6.20.7</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.21 Paired UTF-16 surrogates	
<a href="#">Case 6.21.1</a>	Missing
<a href="#">Case 6.21.2</a>	Missing
<a href="#">Case 6.21.3</a>	Missing
<a href="#">Case 6.21.4</a>	Missing
<a href="#">Case 6.21.5</a>	Missing
<a href="#">Case 6.21.6</a>	Missing
<a href="#">Case 6.21.7</a>	Missing
<a href="#">Case 6.21.8</a>	Missing
6 UTF-8 Handling	dmc_websockets/1.2.0
6.22 Non-character code points (valid UTF-8)	

<a href="#">Case 6.22.1</a>	Missing	
<a href="#">Case 6.22.2</a>	Missing	
<a href="#">Case 6.22.3</a>	Missing	
<a href="#">Case 6.22.4</a>	Missing	
<a href="#">Case 6.22.5</a>	Missing	
<a href="#">Case 6.22.6</a>	Missing	
<a href="#">Case 6.22.7</a>	Missing	
<a href="#">Case 6.22.8</a>	Missing	
<a href="#">Case 6.22.9</a>	Missing	
<a href="#">Case 6.22.10</a>	Missing	
<a href="#">Case 6.22.11</a>	Missing	
<a href="#">Case 6.22.12</a>	Missing	
<a href="#">Case 6.22.13</a>	Missing	
<a href="#">Case 6.22.14</a>	Missing	
<a href="#">Case 6.22.15</a>	Missing	
<a href="#">Case 6.22.16</a>	Missing	
<a href="#">Case 6.22.17</a>	Missing	
<a href="#">Case 6.22.18</a>	Missing	
<a href="#">Case 6.22.19</a>	Missing	
<a href="#">Case 6.22.20</a>	Missing	
<a href="#">Case 6.22.21</a>	Missing	
<a href="#">Case 6.22.22</a>	Missing	
<a href="#">Case 6.22.23</a>	Missing	
<a href="#">Case 6.22.24</a>	Missing	
<a href="#">Case 6.22.25</a>	Missing	
<a href="#">Case 6.22.26</a>	Missing	
<a href="#">Case 6.22.27</a>	Missing	
<a href="#">Case 6.22.28</a>	Missing	
<a href="#">Case 6.22.29</a>	Missing	
<a href="#">Case 6.22.30</a>	Missing	
<a href="#">Case 6.22.31</a>	Missing	
<a href="#">Case 6.22.32</a>	Missing	
<a href="#">Case 6.22.33</a>	Missing	
<a href="#">Case 6.22.34</a>	Missing	
6 UTF-8 Handling	dmc_websockets/1.2.0	
6.23 Unicode specials (i.e. replacement char)		
<a href="#">Case 6.23.1</a>	Missing	
<a href="#">Case 6.23.2</a>	Missing	
<a href="#">Case 6.23.3</a>	Missing	
<a href="#">Case 6.23.4</a>	Missing	
<a href="#">Case 6.23.5</a>	Missing	
<a href="#">Case 6.23.6</a>	Missing	
<a href="#">Case 6.23.7</a>	Missing	
7 Close Handling	dmc_websockets/1.2.0	
7.1 Basic close behavior (fuzzer initiated)		
<a href="#">Case 7.1.1</a>	Pass	1000
<a href="#">Case 7.1.2</a>	Pass	1000
<a href="#">Case 7.1.3</a>	Pass	1000

Case 7.1.4	Pass	1000
Case 7.1.5	Pass	1000
Case 7.1.6	Info	None
7 Close Handling	dmc_websockets/1.2.0	
7.3 Close frame structure: payload length (fuzzer initiated)		
Case 7.3.1	Pass	1000
Case 7.3.2	Pass	1002
Case 7.3.3	Pass	1000
Case 7.3.4	Pass	1000
Case 7.3.5	Pass	1000
Case 7.3.6	Pass	1002
7 Close Handling	dmc_websockets/1.2.0	
7.5 Close frame structure: payload value (fuzzer initiated)		
Case 7.5.1	Missing	
7 Close Handling	dmc_websockets/1.2.0	
7.7 Close frame structure: valid close codes (fuzzer initiated)		
Case 7.7.1	Pass	1000
Case 7.7.2	Pass	1001
Case 7.7.3	Pass	1002
Case 7.7.4	Pass	1003
Case 7.7.5	Pass	1007
Case 7.7.6	Pass	1008
Case 7.7.7	Pass	1009
Case 7.7.8	Pass	1010
Case 7.7.9	Pass	1011
Case 7.7.10	Pass	3000
Case 7.7.11	Pass	3999
Case 7.7.12	Pass	4000
Case 7.7.13	Pass	4999
7 Close Handling	dmc_websockets/1.2.0	
7.9 Close frame structure: invalid close codes (fuzzer initiated)		
Case 7.9.1	Pass	1002
Case 7.9.2	Pass	1002
Case 7.9.3	Pass	1002
Case 7.9.4	Pass	1002
Case 7.9.5	Pass	1002
Case 7.9.6	Pass	1002
Case 7.9.7	Pass	1002
Case 7.9.8	Pass	1002
Case 7.9.9	Pass	1002
Case 7.9.10	Pass	1002
Case 7.9.11	Pass	1002
Case 7.9.12	Pass	1002
Case 7.9.13	Pass	1002
7 Close Handling	dmc_websockets/1.2.0	
7.13 Informational close information (fuzzer initiated)		
Case 7.13.1	Info	5000
Case 7.13.2	Info	65535



9 Limits/Performance		dmc_websockets/1.2.0	
9.1 Text Message (increasing size)			
Case 9.1.1	Pass 201 ms	1000	
Case 9.1.2	Missing		
Case 9.1.3	Missing		
Case 9.1.4	Missing		
Case 9.1.5	Missing		
Case 9.1.6	Missing		
9 Limits/Performance		dmc_websockets/1.2.0	
9.2 Binary Message (increasing size)			
Case 9.2.1	Missing		
Case 9.2.2	Missing		
Case 9.2.3	Missing		
Case 9.2.4	Missing		
Case 9.2.5	Missing		
Case 9.2.6	Missing		
9 Limits/Performance		dmc_websockets/1.2.0	
9.3 Fragmented Text Message (fixed size, increasing fragment size)			
Case 9.3.1	Missing		
Case 9.3.2	Missing		
Case 9.3.3	Missing		
Case 9.3.4	Missing		
Case 9.3.5	Missing		
Case 9.3.6	Missing		
Case 9.3.7	Missing		
Case 9.3.8	Missing		
Case 9.3.9	Missing		
9 Limits/Performance		dmc_websockets/1.2.0	
9.4 Fragmented Binary Message (fixed size, increasing fragment size)			
Case 9.4.1	Missing		
Case 9.4.2	Missing		
Case 9.4.3	Missing		
Case 9.4.4	Missing		
Case 9.4.5	Missing		
Case 9.4.6	Missing		
Case 9.4.7	Missing		
Case 9.4.8	Missing		
Case 9.4.9	Missing		
9 Limits/Performance		dmc_websockets/1.2.0	
9.5 Text Message (fixed size, increasing chop size)			
Case 9.5.1	Missing		
Case 9.5.2	Missing		
Case 9.5.3	Missing		
Case 9.5.4	Missing		
Case 9.5.5	Missing		
Case 9.5.6	Missing		
9 Limits/Performance		dmc_websockets/1.2.0	



9.6 Binary Text Message (fixed size, increasing chop size)		
<a href="#">Case 9.6.1</a>	Missing	
<a href="#">Case 9.6.2</a>	Missing	
<a href="#">Case 9.6.3</a>	Missing	
<a href="#">Case 9.6.4</a>	Missing	
<a href="#">Case 9.6.5</a>	Missing	
<a href="#">Case 9.6.6</a>	Missing	
9 Limits/Performance	dmc_websockets/1.2.0	
9.7 Text Message Roundtrip Time (fixed number, increasing size)		
<a href="#">Case 9.7.1</a>	Missing	
<a href="#">Case 9.7.2</a>	Missing	
<a href="#">Case 9.7.3</a>	Missing	
<a href="#">Case 9.7.4</a>	Missing	
<a href="#">Case 9.7.5</a>	Missing	
<a href="#">Case 9.7.6</a>	Missing	
9 Limits/Performance	dmc_websockets/1.2.0	
9.8 Binary Message Roundtrip Time (fixed number, increasing size)		
<a href="#">Case 9.8.1</a>	Missing	
<a href="#">Case 9.8.2</a>	Missing	
<a href="#">Case 9.8.3</a>	Missing	
<a href="#">Case 9.8.4</a>	Missing	
<a href="#">Case 9.8.5</a>	Missing	
<a href="#">Case 9.8.6</a>	Missing	
10 Misc	dmc_websockets/1.2.0	
10.1 Auto-Fragmentation		
<a href="#">Case 10.1.1</a>	Pass	1000
12 WebSocket Compression (different payloads)	dmc_websockets/1.2.0	
12.1 Large JSON data file (utf8, 194056 bytes)		
<a href="#">Case 12.1.1</a>	Missing	
<a href="#">Case 12.1.2</a>	Missing	
<a href="#">Case 12.1.3</a>	Missing	
<a href="#">Case 12.1.4</a>	Missing	
<a href="#">Case 12.1.5</a>	Missing	
<a href="#">Case 12.1.6</a>	Missing	
<a href="#">Case 12.1.7</a>	Missing	
<a href="#">Case 12.1.8</a>	Missing	
<a href="#">Case 12.1.9</a>	Missing	
<a href="#">Case 12.1.10</a>	Missing	
<a href="#">Case 12.1.11</a>	Missing	
<a href="#">Case 12.1.12</a>	Missing	
<a href="#">Case 12.1.13</a>	Missing	
<a href="#">Case 12.1.14</a>	Missing	
<a href="#">Case 12.1.15</a>	Missing	
<a href="#">Case 12.1.16</a>	Missing	
<a href="#">Case 12.1.17</a>	Missing	
<a href="#">Case 12.1.18</a>	Missing	
12 WebSocket Compression (different payloads)	dmc_websockets/1.2.0	
12.2 Lena Picture, Bitmap 512x512 bw (binary, 263222 bytes)		

<a href="#">Case 12.2.1</a>	Missing
<a href="#">Case 12.2.2</a>	Missing
<a href="#">Case 12.2.3</a>	Missing
<a href="#">Case 12.2.4</a>	Missing
<a href="#">Case 12.2.5</a>	Missing
<a href="#">Case 12.2.6</a>	Missing
<a href="#">Case 12.2.7</a>	Missing
<a href="#">Case 12.2.8</a>	Missing
<a href="#">Case 12.2.9</a>	Missing
<a href="#">Case 12.2.10</a>	Missing
<a href="#">Case 12.2.11</a>	Missing
<a href="#">Case 12.2.12</a>	Missing
<a href="#">Case 12.2.13</a>	Missing
<a href="#">Case 12.2.14</a>	Missing
<a href="#">Case 12.2.15</a>	Missing
<a href="#">Case 12.2.16</a>	Missing
<a href="#">Case 12.2.17</a>	Missing
<a href="#">Case 12.2.18</a>	Missing
12 WebSocket Compression (different payloads)	dmc_websockets/1.2.0
12.3 Human readable text, Goethe's Faust I (German) (binary, 222218 bytes)	
<a href="#">Case 12.3.1</a>	Missing
<a href="#">Case 12.3.2</a>	Missing
<a href="#">Case 12.3.3</a>	Missing
<a href="#">Case 12.3.4</a>	Missing
<a href="#">Case 12.3.5</a>	Missing
<a href="#">Case 12.3.6</a>	Missing
<a href="#">Case 12.3.7</a>	Missing
<a href="#">Case 12.3.8</a>	Missing
<a href="#">Case 12.3.9</a>	Missing
<a href="#">Case 12.3.10</a>	Missing
<a href="#">Case 12.3.11</a>	Missing
<a href="#">Case 12.3.12</a>	Missing
<a href="#">Case 12.3.13</a>	Missing
<a href="#">Case 12.3.14</a>	Missing
<a href="#">Case 12.3.15</a>	Missing
<a href="#">Case 12.3.16</a>	Missing
<a href="#">Case 12.3.17</a>	Missing
<a href="#">Case 12.3.18</a>	Missing
12 WebSocket Compression (different payloads)	dmc_websockets/1.2.0
12.4 Large HTML file (utf8, 263527 bytes)	
<a href="#">Case 12.4.1</a>	Missing
<a href="#">Case 12.4.2</a>	Missing
<a href="#">Case 12.4.3</a>	Missing
<a href="#">Case 12.4.4</a>	Missing
<a href="#">Case 12.4.5</a>	Missing
<a href="#">Case 12.4.6</a>	Missing
<a href="#">Case 12.4.7</a>	Missing
<a href="#">Case 12.4.8</a>	Missing

<a href="#">Case 12.4.9</a>	Missing
<a href="#">Case 12.4.10</a>	Missing
<a href="#">Case 12.4.11</a>	Missing
<a href="#">Case 12.4.12</a>	Missing
<a href="#">Case 12.4.13</a>	Missing
<a href="#">Case 12.4.14</a>	Missing
<a href="#">Case 12.4.15</a>	Missing
<a href="#">Case 12.4.16</a>	Missing
<a href="#">Case 12.4.17</a>	Missing
<a href="#">Case 12.4.18</a>	Missing
12 WebSocket Compression (different payloads)	dmc_websockets/1.2.0
12.5 A larger PDF (binary, 1042328 bytes)	
<a href="#">Case 12.5.1</a>	Missing
<a href="#">Case 12.5.2</a>	Missing
<a href="#">Case 12.5.3</a>	Missing
<a href="#">Case 12.5.4</a>	Missing
<a href="#">Case 12.5.5</a>	Missing
<a href="#">Case 12.5.6</a>	Missing
<a href="#">Case 12.5.7</a>	Missing
<a href="#">Case 12.5.8</a>	Missing
<a href="#">Case 12.5.9</a>	Missing
<a href="#">Case 12.5.10</a>	Missing
<a href="#">Case 12.5.11</a>	Missing
<a href="#">Case 12.5.12</a>	Missing
<a href="#">Case 12.5.13</a>	Missing
<a href="#">Case 12.5.14</a>	Missing
<a href="#">Case 12.5.15</a>	Missing
<a href="#">Case 12.5.16</a>	Missing
<a href="#">Case 12.5.17</a>	Missing
<a href="#">Case 12.5.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.1 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]	
<a href="#">Case 13.1.1</a>	Missing
<a href="#">Case 13.1.2</a>	Missing
<a href="#">Case 13.1.3</a>	Missing
<a href="#">Case 13.1.4</a>	Missing
<a href="#">Case 13.1.5</a>	Missing
<a href="#">Case 13.1.6</a>	Missing
<a href="#">Case 13.1.7</a>	Missing
<a href="#">Case 13.1.8</a>	Missing
<a href="#">Case 13.1.9</a>	Missing
<a href="#">Case 13.1.10</a>	Missing
<a href="#">Case 13.1.11</a>	Missing
<a href="#">Case 13.1.12</a>	Missing
<a href="#">Case 13.1.13</a>	Missing
<a href="#">Case 13.1.14</a>	Missing
<a href="#">Case 13.1.15</a>	Missing
<a href="#">Case 13.1.16</a>	Missing

<a href="#">Case 13.1.17</a>	Missing
<a href="#">Case 13.1.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.2 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]	
<a href="#">Case 13.2.1</a>	Missing
<a href="#">Case 13.2.2</a>	Missing
<a href="#">Case 13.2.3</a>	Missing
<a href="#">Case 13.2.4</a>	Missing
<a href="#">Case 13.2.5</a>	Missing
<a href="#">Case 13.2.6</a>	Missing
<a href="#">Case 13.2.7</a>	Missing
<a href="#">Case 13.2.8</a>	Missing
<a href="#">Case 13.2.9</a>	Missing
<a href="#">Case 13.2.10</a>	Missing
<a href="#">Case 13.2.11</a>	Missing
<a href="#">Case 13.2.12</a>	Missing
<a href="#">Case 13.2.13</a>	Missing
<a href="#">Case 13.2.14</a>	Missing
<a href="#">Case 13.2.15</a>	Missing
<a href="#">Case 13.2.16</a>	Missing
<a href="#">Case 13.2.17</a>	Missing
<a href="#">Case 13.2.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.3 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]	
<a href="#">Case 13.3.1</a>	Missing
<a href="#">Case 13.3.2</a>	Missing
<a href="#">Case 13.3.3</a>	Missing
<a href="#">Case 13.3.4</a>	Missing
<a href="#">Case 13.3.5</a>	Missing
<a href="#">Case 13.3.6</a>	Missing
<a href="#">Case 13.3.7</a>	Missing
<a href="#">Case 13.3.8</a>	Missing
<a href="#">Case 13.3.9</a>	Missing
<a href="#">Case 13.3.10</a>	Missing
<a href="#">Case 13.3.11</a>	Missing
<a href="#">Case 13.3.12</a>	Missing
<a href="#">Case 13.3.13</a>	Missing
<a href="#">Case 13.3.14</a>	Missing
<a href="#">Case 13.3.15</a>	Missing
<a href="#">Case 13.3.16</a>	Missing
<a href="#">Case 13.3.17</a>	Missing
<a href="#">Case 13.3.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.4 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]	
<a href="#">Case 13.4.1</a>	Missing
<a href="#">Case 13.4.2</a>	Missing

<a href="#">Case 13.4.3</a>	Missing
<a href="#">Case 13.4.4</a>	Missing
<a href="#">Case 13.4.5</a>	Missing
<a href="#">Case 13.4.6</a>	Missing
<a href="#">Case 13.4.7</a>	Missing
<a href="#">Case 13.4.8</a>	Missing
<a href="#">Case 13.4.9</a>	Missing
<a href="#">Case 13.4.10</a>	Missing
<a href="#">Case 13.4.11</a>	Missing
<a href="#">Case 13.4.12</a>	Missing
<a href="#">Case 13.4.13</a>	Missing
<a href="#">Case 13.4.14</a>	Missing
<a href="#">Case 13.4.15</a>	Missing
<a href="#">Case 13.4.16</a>	Missing
<a href="#">Case 13.4.17</a>	Missing
<a href="#">Case 13.4.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.5 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<a href="#">Case 13.5.1</a>	Missing
<a href="#">Case 13.5.2</a>	Missing
<a href="#">Case 13.5.3</a>	Missing
<a href="#">Case 13.5.4</a>	Missing
<a href="#">Case 13.5.5</a>	Missing
<a href="#">Case 13.5.6</a>	Missing
<a href="#">Case 13.5.7</a>	Missing
<a href="#">Case 13.5.8</a>	Missing
<a href="#">Case 13.5.9</a>	Missing
<a href="#">Case 13.5.10</a>	Missing
<a href="#">Case 13.5.11</a>	Missing
<a href="#">Case 13.5.12</a>	Missing
<a href="#">Case 13.5.13</a>	Missing
<a href="#">Case 13.5.14</a>	Missing
<a href="#">Case 13.5.15</a>	Missing
<a href="#">Case 13.5.16</a>	Missing
<a href="#">Case 13.5.17</a>	Missing
<a href="#">Case 13.5.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.6 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]	
<a href="#">Case 13.6.1</a>	Missing
<a href="#">Case 13.6.2</a>	Missing
<a href="#">Case 13.6.3</a>	Missing
<a href="#">Case 13.6.4</a>	Missing
<a href="#">Case 13.6.5</a>	Missing
<a href="#">Case 13.6.6</a>	Missing
<a href="#">Case 13.6.7</a>	Missing
<a href="#">Case 13.6.8</a>	Missing
<a href="#">Case 13.6.9</a>	Missing

<a href="#">Case 13.6.10</a>	Missing
<a href="#">Case 13.6.11</a>	Missing
<a href="#">Case 13.6.12</a>	Missing
<a href="#">Case 13.6.13</a>	Missing
<a href="#">Case 13.6.14</a>	Missing
<a href="#">Case 13.6.15</a>	Missing
<a href="#">Case 13.6.16</a>	Missing
<a href="#">Case 13.6.17</a>	Missing
<a href="#">Case 13.6.18</a>	Missing
13 WebSocket Compression (different parameters)	dmc_websockets/1.2.0
13.7 Large JSON data file (utf8, 194056 bytes) - client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)] / server accept (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<a href="#">Case 13.7.1</a>	Missing
<a href="#">Case 13.7.2</a>	Missing
<a href="#">Case 13.7.3</a>	Missing
<a href="#">Case 13.7.4</a>	Missing
<a href="#">Case 13.7.5</a>	Missing
<a href="#">Case 13.7.6</a>	Missing
<a href="#">Case 13.7.7</a>	Missing
<a href="#">Case 13.7.8</a>	Missing
<a href="#">Case 13.7.9</a>	Missing
<a href="#">Case 13.7.10</a>	Missing
<a href="#">Case 13.7.11</a>	Missing
<a href="#">Case 13.7.12</a>	Missing
<a href="#">Case 13.7.13</a>	Missing
<a href="#">Case 13.7.14</a>	Missing
<a href="#">Case 13.7.15</a>	Missing
<a href="#">Case 13.7.16</a>	Missing
<a href="#">Case 13.7.17</a>	Missing
<a href="#">Case 13.7.18</a>	Missing

## Case 1.1.1

Case Description

Send text message with payload 0.

Up

Case Expectation

Receive echo'ed text message (with empty payload). Clean close with normal code.

## Case 1.1.2

Case Description

Send text message message with payload of length 125.

Up

**Case Expectation**

Receive echo'ed text message (with payload as sent). Clean close with normal code.

**Case 1.1.3**

**Case Description**

Up

Send text message message with payload of length 126.

**Case Expectation**

Receive echo'ed text message (with payload as sent). Clean close with normal code.

**Case 1.1.4**

**Case Description**

Up

Send text message message with payload of length 127.

**Case Expectation**

Receive echo'ed text message (with payload as sent). Clean close with normal code.

**Case 1.1.5**

**Case Description**

Up

Send text message message with payload of length 128.

**Case Expectation**

Receive echo'ed text message (with payload as sent). Clean close with normal code.

**Case 1.1.6**

**Case Description**

Up

Send text message message with payload of length 65535.

**Case Expectation**

Receive echo'ed text message (with payload as sent). Clean close with normal code.

**Case 1.1.7**

**Case Description**

Up

Send text message message with payload of length 65536.

**Case Expectation**



Receive echo'ed text message (with payload as sent). Clean close with normal code.

## Case 1.1.8

Case Description

Send text message message with payload of length 65536. Sent out data in chops of 997 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent). Clean close with normal code.

## Case 1.2.1

Case Description

Send binary message with payload 0.

Up

Case Expectation

Receive echo'ed binary message (with empty payload). Clean close with normal code.

## Case 1.2.2

Case Description

Send binary message message with payload of length 125.

Up

Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.3

Case Description

Send binary message message with payload of length 126.

Up

Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.4

Case Description

Send binary message message with payload of length 127.

Up

Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.5

### Case Description

Up

Send binary message message with payload of length 128.

### Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.6

### Case Description

Up

Send binary message message with payload of length 65535.

### Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.7

### Case Description

Up

Send binary message message with payload of length 65536.

### Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 1.2.8

### Case Description

Up

Send binary message message with payload of length 65536. Sent out data in chops of 997 octets.

### Case Expectation

Receive echo'ed binary message (with payload as sent). Clean close with normal code.

## Case 2.1

### Case Description

Up

Send ping without payload.

### Case Expectation

Pong (with empty payload) is sent in reply to Ping. Clean close with normal code.

## Case 2.2

<div><div>Case Description</div><div>Send ping with small text payload.</div></div>	Up
<div><div>Case Expectation</div><div>Pong with payload echo'ed is sent in reply to Ping. Clean close with normal code.</div></div>	

Case 2.3

<div><div>Case Description</div><div>Send ping with small binary (non UTF-8) payload.</div></div>	Up
<div><div>Case Expectation</div><div>Pong with payload echo'ed is sent in reply to Ping. Clean close with normal code.</div></div>	

Case 2.4

<div><div>Case Description</div><div>Send ping with binary payload of 125 octets.</div></div>	Up
<div><div>Case Expectation</div><div>Pong with payload echo'ed is sent in reply to Ping. Clean close with normal code.</div></div>	

Case 2.5

<div><div>Case Description</div><div>Send ping with binary payload of 126 octets.</div></div>	Up
<div><div>Case Expectation</div><div>Connection is failed immediately (1002/Protocol Error), since control frames are only allowed to have payload up to and including 125 octets..</div></div>	

Case 2.6

<div><div>Case Description</div><div>Send ping with binary payload of 125 octets, send in octet-wise chops.</div></div>	Up
<div><div>Case Expectation</div><div>Pong with payload echo'ed is sent in reply to Ping. Implementations must be TCP clean. Clean close with normal code.</div></div>	

Case 2.7

<div><div>Case Description</div><div></div></div>	Up
---	----

Send unsolicited pong without payload. Verify nothing is received. Clean close with normal code.

Case Expectation

Nothing.

## Case 2.8

Case Description

Send unsolicited pong with payload. Verify nothing is received. Clean close with normal code.

Up

Case Expectation

Nothing.

## Case 2.9

Case Description

Send unsolicited pong with payload. Send ping with payload. Verify pong for ping is received.

Up

Case Expectation

Nothing in reply to own Pong, but Pong with payload echo'ed in reply to Ping. Clean close with normal code.

## Case 2.10

Case Description

Send 10 Pings with payload.

Up

Case Expectation

Pongs for our Pings with all the payloads. Note: This is not required by the Spec .. but we check for this behaviour anyway. Clean close with normal code.

## Case 2.11

Case Description

Send 10 Pings with payload. Send out octets in octet-wise chops.

Up

Case Expectation

Pongs for our Pings with all the payloads. Note: This is not required by the Spec .. but we check for this behaviour anyway. Clean close with normal code.

## Case 3.1

Case Description

Send small text message with **RSV = 1**.

Up

Case Expectation

The connection is failed immediately (1002/protocol error), since RSV must be 0, when no extension defining RSV meaning has been negotiated.

Case 3.2

Case Description

Up

Send small text message, then send again with **RSV = 2**, then send Ping.

Case Expectation

Echo for first message is received, but then connection is failed immediately, since RSV must be 0, when no extension defining RSV meaning has been negotiated. The Pong is not received.

Case 3.3

Case Description

Up

Send small text message, then send again with **RSV = 3**, then send Ping. Octets are sent in frame-wise chops. Octets are sent in octet-wise chops.

Case Expectation

Echo for first message is received, but then connection is failed immediately, since RSV must be 0, when no extension defining RSV meaning has been negotiated. The Pong is not received.

Case 3.4

Case Description

Up

Send small text message, then send again with **RSV = 4**, then send Ping. Octets are sent in octet-wise chops.

Case Expectation

Echo for first message is received, but then connection is failed immediately, since RSV must be 0, when no extension defining RSV meaning has been negotiated. The Pong is not received.

Case 3.5

Case Description

Up

Send small binary message with **RSV = 5**.

Case Expectation

The connection is failed immediately, since RSV must be 0.

Case 3.6

Case Description

Up

Send Ping with **RSV = 6**.

Case Expectation

The connection is failed immediately, since RSV must be 0.

Case 3.7

Case Description

Up

Send Close with **RSV = 7**.

Case Expectation

The connection is failed immediately, since RSV must be 0.

Case 4.1.1

Case Description

Up

Send frame with reserved non-control **Opcode = 3**.

Case Expectation

The connection is failed immediately.

Case 4.1.2

Case Description

Up

Send frame with reserved non-control **Opcode = 4** and non-empty payload.

Case Expectation

The connection is failed immediately.

Case 4.1.3

Case Description

Up

Send small text message, then send frame with reserved non-control **Opcode = 5**, then send Ping.

Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.

Case 4.1.4

Case Description

Up

Send small text message, then send frame with reserved non-control **Opcode = 6** and non-empty payload, then send Ping.

Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.

## Case 4.1.5

### Case Description

Up

Send small text message, then send frame with reserved non-control **Opcode = 7** and non-empty payload, then send Ping.

### Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.

## Case 4.2.1

### Case Description

Up

Send frame with reserved control **Opcode = 11**.

### Case Expectation

The connection is failed immediately.

## Case 4.2.2

### Case Description

Up

Send frame with reserved control **Opcode = 12** and non-empty payload.

### Case Expectation

The connection is failed immediately.

## Case 4.2.3

### Case Description

Up

Send small text message, then send frame with reserved control **Opcode = 13**, then send Ping.

### Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.

## Case 4.2.4

### Case Description

Up

Send small text message, then send frame with reserved control **Opcode = 14** and non-empty payload, then send Ping.

### Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.



## Case 4.2.5

### Case Description

Up

Send small text message, then send frame with reserved control **Opcode = 15** and non-empty payload, then send Ping.

### Case Expectation

Echo for first message is received, but then connection is failed immediately, since reserved opcode frame is used. A Pong is not received.

## Case 5.1

### Case Description

Up

Send Ping fragmented into 2 fragments.

### Case Expectation

Connection is failed immediately, since control message MUST NOT be fragmented.

## Case 5.2

### Case Description

Up

Send Pong fragmented into 2 fragments.

### Case Expectation

Connection is failed immediately, since control message MUST NOT be fragmented.

## Case 5.3

### Case Description

Up

Send text Message fragmented into 2 fragments.

### Case Expectation

Message is processed and echo'ed back to us.

## Case 5.4

### Case Description

Up

Send text Message fragmented into 2 fragments, octets are sent in frame-wise chops.

### Case Expectation

Message is processed and echo'ed back to us.

## Case 5.5

<b>Case Description</b>	Up
Send text Message fragmented into 2 fragments, octets are sent in octet-wise chops.	
<b>Case Expectation</b>	
Message is processed and echo'ed back to us.	

Case 5.6

<b>Case Description</b>	Up
Send text Message fragmented into 2 fragments, one ping with payload in-between.	
<b>Case Expectation</b>	
A pong is received, then the message is echo'ed back to us.	

Case 5.7

<b>Case Description</b>	Up
Send text Message fragmented into 2 fragments, one ping with payload in-between. Octets are sent in frame-wise chops.	
<b>Case Expectation</b>	
A pong is received, then the message is echo'ed back to us.	

Case 5.8

<b>Case Description</b>	Up
Send text Message fragmented into 2 fragments, one ping with payload in-between. Octets are sent in octet-wise chops.	
<b>Case Expectation</b>	
A pong is received, then the message is echo'ed back to us.	

Case 5.9

<b>Case Description</b>	Up
Send unfragmented Text Message after Continuation Frame with FIN = true, where there is nothing to continue, sent in one chop.	
<b>Case Expectation</b>	
The connection is failed immediately, since there is no message to continue.	

Case 5.10

<b>Case Description</b>	Up
-------------------------	----

Send unfragmented Text Message after Continuation Frame with FIN = true, where there is nothing to continue, sent in per-frame chops.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

## Case 5.11

**Case Description**

Up

Send unfragmented Text Message after Continuation Frame with FIN = true, where there is nothing to continue, sent in octet-wise chops.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

## Case 5.12

**Case Description**

Up

Send unfragmented Text Message after Continuation Frame with FIN = false, where there is nothing to continue, sent in one chop.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

## Case 5.13

**Case Description**

Up

Send unfragmented Text Message after Continuation Frame with FIN = false, where there is nothing to continue, sent in per-frame chops.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

## Case 5.14

**Case Description**

Up

Send unfragmented Text Message after Continuation Frame with FIN = false, where there is nothing to continue, sent in octet-wise chops.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

## Case 5.15

**Case Description**

Up

Send text Message fragmented into 2 fragments, then Continuation Frame with FIN = false where there is nothing to continue, then unfragmented Text Message, all sent in one chop.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

**Case 5.16**

**Case Description**

Up

Repeated 2x: Continuation Frame with FIN = false (where there is nothing to continue), then text Message fragmented into 2 fragments.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

**Case 5.17**

**Case Description**

Up

Repeated 2x: Continuation Frame with FIN = true (where there is nothing to continue), then text Message fragmented into 2 fragments.

**Case Expectation**

The connection is failed immediately, since there is no message to continue.

**Case 5.18**

**Case Description**

Up

Send text Message fragmented into 2 fragments, with both frame opcodes set to text, sent in one chop.

**Case Expectation**

The connection is failed immediately, since all data frames after the initial data frame must have opcode 0.

**Case 5.19**

**Case Description**

Up

A fragmented text message is sent in multiple frames. After sending the first 2 frames of the text message, a Ping is sent. Then we wait 1s, then we send 2 more text fragments, another Ping and then the final text fragment. Everything is legal.

**Case Expectation**

The peer immediately answers the first Ping before it has received the last text message fragment. The peer pong's back the Ping's payload exactly, and echo's the payload of the fragmented message back to us.

**Case 5.20**

**Case Description**

Up

Same as Case 5.19, but send all frames with SYNC = True. Note, this does not change the octets sent in any way, only how the stream is chopped up on the wire.

Case Expectation

Same as Case 5.19. Implementations must be agnostic to how octet stream is chopped up on wire (must be TCP clean).

Case 6.1.1

Case Description

Up

Send text message of length 0.

Case Expectation

A message is echo'ed back to us (with empty payload).

Case 6.1.2

Case Description

Up

Send fragmented text message, 3 fragments each of length 0.

Case Expectation

A message is echo'ed back to us (with empty payload).

Case 6.1.3

Case Description

Up

Send fragmented text message, 3 fragments, first and last of length 0, middle non-empty.

Case Expectation

A message is echo'ed back to us (with payload = payload of middle fragment).

Case 6.2.1

Case Description

Up

Send a valid UTF-8 text message in one fragment.

MESSAGE:  
Hello-μ@ßöäüää-UTF-8!!  
48656c6c6f2dc2b540c39fc3b6c3a4c3bcc3a0c3a12d5554462d382121

Case Expectation

The message is echo'ed back to us.

Case 6.2.2

Case Description

Up

Send a valid UTF-8 text message in two fragments, fragmented on UTF-8 code point boundary.

MESSAGE FRAGMENT 1:  
Hello-μ@βöä  
48656c6c6f2dc2b540c39fc3b6c3a4

MESSAGE FRAGMENT 2:  
üää-UTF-8!!  
c3bcc3a0c3a12d5554462d382121

Case Expectation

The message is echo'ed back to us.

Case 6.2.3

Case Description

Up

Send a valid UTF-8 text message in fragments of 1 octet, resulting in frames ending on positions which are not code point ends.

MESSAGE:  
Hello-μ@βöäüää-UTF-8!!  
48656c6c6f2dc2b540c39fc3b6c3a4c3bcc3a0c3a12d5554462d382121

Case Expectation

The message is echo'ed back to us.

Case 6.2.4

Case Description

Up

Send a valid UTF-8 text message in fragments of 1 octet, resulting in frames ending on positions which are not code point ends.

MESSAGE:  
κόσμε  
cebae1bdb9cf83cebcecb5

Case Expectation

The message is echo'ed back to us.

Case 6.3.1

Case Description

Up

Send invalid UTF-8 text message unfragmented.

MESSAGE:  
cebae1bdb9cf83cebcecb5eda080656469746564

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.3.2

Case Description

Up

Send invalid UTF-8 text message in fragments of 1 octet, resulting in frames ending on positions which are not code point ends.

MESSAGE:  
cebae1bdb9cf83cebcb5eda080656469746564

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.4.1

Case Description

Up

Send invalid UTF-8 text message in 3 fragments (frames). First frame payload is valid, then wait, then 2nd frame which contains the payload making the sequence invalid, then wait, then 3rd frame with rest. Note that PART1 and PART3 are valid UTF-8 in themselves, PART2 is a 0x110000 encoded as in the UTF-8 integer encoding scheme, but the codepoint is invalid (out of range).

MESSAGE PARTS:  
PART1 = ceba1bdb9cf83cebcb5  
PART2 = f4908080  
PART3 = 656469746564

Case Expectation

The first frame is accepted, we expect to timeout on the first wait. The 2nd frame should be rejected immediately (fail fast on UTF-8). If we timeout, we expect the connection is failed at least then, since the complete message payload is not valid UTF-8.

Case 6.4.2

Case Description

Up

Same as Case 6.4.1, but in 2nd frame, we send only up to and including the octet making the complete payload invalid.

MESSAGE PARTS:  
PART1 = ceba1bdb9cf83cebcb5f4  
PART2 = 90  
PART3 = 8080656469746564

Case Expectation

The first frame is accepted, we expect to timeout on the first wait. The 2nd frame should be rejected immediately (fail fast on UTF-8). If we timeout, we expect the connection is failed at least then, since the complete message payload is not valid UTF-8.

Case 6.4.3

Case Description

Up

Same as Case 6.4.1, but we send message not in 3 frames, but in 3 chops of the same message frame.

MESSAGE PARTS:  
PART1 = ceba1bdb9cf83cebcb5  
PART2 = f4908080  
PART3 = 656469746564

Case Expectation

The first chop is accepted, we expect to timeout on the first wait. The 2nd chop should be rejected immediately (fail fast on UTF-8). If we timeout, we expect the connection is failed at least then, since the complete message payload is not valid UTF-8.



## Case 6.4.4

Case Description

Up

Same as Case 6.4.2, but we send message not in 3 frames, but in 3 chops of the same message frame.

MESSAGE PARTS:  
PART1 = cebae1bdb9cf83cebcceb5f4  
PART2 = 90  
PART3 =

Case Expectation

The first chop is accepted, we expect to timeout on the first wait. The 2nd chop should be rejected immediately (fail fast on UTF-8). If we timeout, we expect the connection is failed at least then, since the complete message payload is not valid UTF-8.

## Case 6.5.1

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x68656c6c6f24776f726c64

Case Expectation

The message is echo'ed back to us.

## Case 6.5.2

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x68656c6c6fc2a2776f726c64

Case Expectation

The message is echo'ed back to us.

## Case 6.5.3

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x68656c6c6fe282ac776f726c64

Case Expectation

The message is echo'ed back to us.

## Case 6.5.4

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x68656c6c6ff0a4ada2776f726c64

**Case Expectation**

The message is echo'ed back to us.

**Case 6.5.5**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xcebae1bdb9cf83cebcceb5

**Case Expectation**

The message is echo'ed back to us.

**Case 6.6.1**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xce

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.6.2**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xceba

**Case Expectation**

The message is echo'ed back to us.

**Case 6.6.3**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xcebae1

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.6.4

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xcebae1bd

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.6.5

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.  
  
Payload: 0xcebae1bdb9

Case Expectation

The message is echo'ed back to us.

## Case 6.6.6

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xcebae1bdb9cf

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.6.7

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.  
  
Payload: 0xcebae1bdb9cf83

Case Expectation

The message is echo'ed back to us.

## Case 6.6.8

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xcebae1bdb9cf83ce

Case Expectation

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.6.9**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xcebae1bdb9cf83cebc

**Case Expectation**

The message is echo'ed back to us.

**Case 6.6.10**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xcebae1bdb9cf83cebccc

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.6.11**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xcebae1bdb9cf83cebccc5

**Case Expectation**

The message is echo'ed back to us.

**Case 6.7.1**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x00

**Case Expectation**

The message is echo'ed back to us.

**Case 6.7.2**

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is valid UTF-8 in one fragment.</div><div>Payload: 0xc280</div></div>
<div><div><b>Case Expectation</b></div><div>The message is echo'ed back to us.</div></div>

Case 6.7.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is valid UTF-8 in one fragment.</div><div>Payload: 0xe0a080</div></div>
<div><div><b>Case Expectation</b></div><div>The message is echo'ed back to us.</div></div>

Case 6.7.4

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is valid UTF-8 in one fragment.</div><div>Payload: 0xf0908080</div></div>
<div><div><b>Case Expectation</b></div><div>The message is echo'ed back to us.</div></div>

Case 6.8.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf888808080</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.8.2

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xfc8480808080</div></div>
<div><div><b>Case Expectation</b></div></div>

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.9.1

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0x7f

**Case Expectation**

The message is echo'ed back to us.

## Case 6.9.2

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xdfbf

**Case Expectation**

The message is echo'ed back to us.

## Case 6.9.3

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfbf

**Case Expectation**

The message is echo'ed back to us.

## Case 6.9.4

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf48fbfbf

**Case Expectation**

The message is echo'ed back to us.

## Case 6.10.1

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment

Payload: 0xf7bfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.10.2**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xfbbfbfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.10.3**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xfdbfbfbfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.11.1**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xed9fbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.11.2**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xee8080

**Case Expectation**

The message is echo'ed back to us.



### Case 6.11.3

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfbdb

Case Expectation

The message is echo'ed back to us.

### Case 6.11.4

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf48fbfbf

Case Expectation

The message is echo'ed back to us.

### Case 6.11.5

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xf4908080

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

### Case 6.12.1

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0x80

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

### Case 6.12.2

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.12.3**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0x80bf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.12.4**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0x80bf80

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.12.5**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0x80bf80bf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.12.6**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0x80bf80bf80

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.12.7**

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0x80bf80bf80bf</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.12.8

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0x808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbe</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.13.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xc020c120c220c320c420c520c620c720c820c920ca20cb20cc20cd20ce20cf20d020d120d220d320d420d520d620d720d820d920da20db20dc20dd20de20</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.13.2

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xe020e120e220e320e420e520e620e720e820e920ea20eb20ec20ed20ee20</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.13.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf020f120f220f320f420f520f620</div></div>
---

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.13.4**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xf820f920fa20

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.13.5**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xfc20

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.14.1**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xc0

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.14.2**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xe080

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.14.3**

<div><div>Case Description</div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf08080</div></div>	Up
<div><div>Case Expectation</div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>	

Case 6.14.4

<div><div>Case Description</div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf8808080</div></div>	Up
<div><div>Case Expectation</div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>	

Case 6.14.5

<div><div>Case Description</div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xfc80808080</div></div>	Up
<div><div>Case Expectation</div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>	

Case 6.14.6

<div><div>Case Description</div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xdf</div></div>	Up
<div><div>Case Expectation</div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>	

Case 6.14.7

<div><div>Case Description</div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xefbf</div></div>	Up
<div><div>Case Expectation</div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>	

## Case 6.14.8

<b>Case Description</b>	Up
Send a text message with payload which is not valid UTF-8 in one fragment.	
Payload: 0xf7bfbf	
<b>Case Expectation</b>	
The connection is failed immediately, since the payload is not valid UTF-8.	

## Case 6.14.9

<b>Case Description</b>	Up
Send a text message with payload which is not valid UTF-8 in one fragment.	
Payload: 0xfbbfbfbf	
<b>Case Expectation</b>	
The connection is failed immediately, since the payload is not valid UTF-8.	

## Case 6.14.10

<b>Case Description</b>	Up
Send a text message with payload which is not valid UTF-8 in one fragment.	
Payload: 0xfdbfbfbfbf	
<b>Case Expectation</b>	
The connection is failed immediately, since the payload is not valid UTF-8.	

## Case 6.15.1

<b>Case Description</b>	Up
Send a text message with payload which is not valid UTF-8 in one fragment.	
Payload: 0xc0e080f08080f8808080fc80808080dfefbff7bfbffbbfbfbffdbfbfbfbf	
<b>Case Expectation</b>	
The connection is failed immediately, since the payload is not valid UTF-8.	

## Case 6.16.1

<b>Case Description</b>	Up
Send a text message with payload which is not valid UTF-8 in one fragment.	

Payload: 0xfe
<b>Case Expectation</b>  The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.16.2

<b>Case Description</b> <div>Up</div> Send a text message with payload which is not valid UTF-8 in one fragment.  Payload: 0xff
<b>Case Expectation</b>  The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.16.3

<b>Case Description</b> <div>Up</div> Send a text message with payload which is not valid UTF-8 in one fragment.  Payload: 0xfefeffff
<b>Case Expectation</b>  The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.17.1

<b>Case Description</b> <div>Up</div> Send a text message with payload which is not valid UTF-8 in one fragment.  Payload: 0xc0af
<b>Case Expectation</b>  The connection is failed immediately, since the payload is not valid UTF-8.

Case 6.17.2

<b>Case Description</b> <div>Up</div> Send a text message with payload which is not valid UTF-8 in one fragment.  Payload: 0xe080af
<b>Case Expectation</b>  The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.17.3

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xf08080af

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.17.4

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xf8808080af

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.17.5

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xfc80808080af

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.18.1

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xc1bf

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.18.2

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.  
  
Payload: 0xe09fbf

Case Expectation



**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.18.3**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xf08fbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.18.4**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xf887bfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.18.5**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xfc83bfbfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.19.1**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xc080

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.19.2**

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xe08080</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.19.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf0808080</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.19.4

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xf880808080</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.19.5

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xfc8080808080</div></div>
<div><div><b>Case Expectation</b></div><div>The connection is failed immediately, since the payload is not valid UTF-8.</div></div>

Case 6.20.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send a text message with payload which is not valid UTF-8 in one fragment.</div><div>Payload: 0xeda080</div></div>
<div><div><b>Case Expectation</b></div></div>

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.20.2

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedadbfbf

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.20.3

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedae80

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.20.4

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedafbfbf

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.20.5

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedb080

Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.20.6

Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment

Payload: 0xedbe80

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.20.7**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.21.1**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xeda080edb080

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.21.2**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xeda080edbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.21.3**

**Case Description**

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedadbfedb080

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.21.4

### Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedadbfbfbf

### Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.21.5

### Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedae80edb080

### Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.21.6

### Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedae80edbfbf

### Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.21.7

### Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedaafbfbfdb080

### Case Expectation

The connection is failed immediately, since the payload is not valid UTF-8.

## Case 6.21.8

### Case Description

Up

Send a text message with payload which is not valid UTF-8 in one fragment.

Payload: 0xedaafbfbfbfbf

**Case Expectation**

The connection is failed immediately, since the payload is not valid UTF-8.

**Case 6.22.1**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.2**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.3**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf09fbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.4**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf09fbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.5**

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf0afbfbfe</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.6

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf0afbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.7

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf0bfbfbfe</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.8

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf0bfbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.9

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf18fbfbfe</div></div></div>
<div><div>Case Expectation</div></div>

The message is echo'ed back to us.

## Case 6.22.10

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf18fbfbf

Case Expectation

The message is echo'ed back to us.

## Case 6.22.11

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf19fbfbe

Case Expectation

The message is echo'ed back to us.

## Case 6.22.12

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf19fbfbf

Case Expectation

The message is echo'ed back to us.

## Case 6.22.13

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf1afbabe

Case Expectation

The message is echo'ed back to us.

## Case 6.22.14

Case Description

Up



Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf1afbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.15**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf1bfbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.16**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf1bfbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.17**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf28fbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.18**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf28fbfbf

**Case Expectation**

The message is echo'ed back to us.

## Case 6.22.19

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf29fbfbe

Case Expectation

The message is echo'ed back to us.

## Case 6.22.20

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf29fbfbf

Case Expectation

The message is echo'ed back to us.

## Case 6.22.21

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf2afbfbf

Case Expectation

The message is echo'ed back to us.

## Case 6.22.22

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf2afbfbf

Case Expectation

The message is echo'ed back to us.

## Case 6.22.23

Case Description

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf2bfbfbe

Case Expectation

The message is echo'ed back to us.

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.24**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf2bfbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.25**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf38fbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.26**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf38fbfbf

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.27**

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf39fbfbe

**Case Expectation**

The message is echo'ed back to us.

**Case 6.22.28**

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf39fbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.29

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf3afbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.30

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf3afbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.31

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf3bfbfbf</div></div></div>
<div><div>Case Expectation</div><div>The message is echo'ed back to us.</div></div>

Case 6.22.32

<div><div>Up</div><div><div>Case Description</div><div>Send a text message with payload which is valid UTF-8 in one fragment.  Payload: 0xf3bfbfbf</div></div></div>
<div><div>Case Expectation</div></div>

The message is echo'ed back to us.

## Case 6.22.33

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf48fbfbe

**Case Expectation**

The message is echo'ed back to us.

## Case 6.22.34

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xf48fbfbf

**Case Expectation**

The message is echo'ed back to us.

## Case 6.23.1

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb9

**Case Expectation**

The message is echo'ed back to us.

## Case 6.23.2

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfba

**Case Expectation**

The message is echo'ed back to us.

## Case 6.23.3

**Case Description**

Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb**b**

**Case Expectation**

The message is echo'ed back to us.

Case 6.23.4

**Case Description** Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb**c**

**Case Expectation**

The message is echo'ed back to us.

Case 6.23.5

**Case Description** Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb**d**

**Case Expectation**

The message is echo'ed back to us.

Case 6.23.6

**Case Description** Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb**e**

**Case Expectation**

The message is echo'ed back to us.

Case 6.23.7

**Case Description** Up

Send a text message with payload which is valid UTF-8 in one fragment.

Payload: 0xefbfb**f**

**Case Expectation**

The message is echo'ed back to us.

## Case 7.1.1

<b>Case Description</b>	Up
Send a message followed by a close frame	
<b>Case Expectation</b>	
Echoed message followed by clean close with normal code.	

## Case 7.1.2

<b>Case Description</b>	Up
Send two close frames	
<b>Case Expectation</b>	
Clean close with normal code. Second close frame ignored.	

## Case 7.1.3

<b>Case Description</b>	Up
Send a ping after close message	
<b>Case Expectation</b>	
Clean close with normal code, no pong.	

## Case 7.1.4

<b>Case Description</b>	Up
Send text message after sending a close frame.	
<b>Case Expectation</b>	
Clean close with normal code. Text message ignored.	

## Case 7.1.5

<b>Case Description</b>	Up
Send message fragment1 followed by close then fragment	
<b>Case Expectation</b>	
Clean close with normal code.	

## Case 7.1.6

<div><div>Up</div><div><b>Case Description</b></div><div>Send 256K message followed by close then a ping</div></div>
<div><div><b>Case Expectation</b></div><div>Case outcome depends on implimentation defined close behavior. Message and close frame are sent back to back. If the close frame is processed before the text message write is complete (as can happen in asynchronous processing models) the close frame is processed first and the text message may not be recieved or may only be partially recieved.</div></div>

Case 7.3.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send a close frame with payload length 0 (no close code, no close reason)</div></div>
<div><div><b>Case Expectation</b></div><div>Clean close with normal code.</div></div>

Case 7.3.2

<div><div>Up</div><div><b>Case Description</b></div><div>Send a close frame with payload length 1</div></div>
<div><div><b>Case Expectation</b></div><div>Clean close with protocol error or drop TCP.</div></div>

Case 7.3.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send a close frame with payload length 2 (regular close with a code)</div></div>
<div><div><b>Case Expectation</b></div><div>Clean close with normal code.</div></div>

Case 7.3.4

<div><div>Up</div><div><b>Case Description</b></div><div>Send a close frame with close code and close reason</div></div>
<div><div><b>Case Expectation</b></div><div>Clean close with normal code.</div></div>

Case 7.3.5

<div><div>Up</div><div><b>Case Description</b></div><div>Send a close frame with payload length 2 (regular close with a code)</div></div>
<div><div><b>Case Expectation</b></div><div>Clean close with normal code.</div></div>



Case Description

Send a close frame with close code and close reason of maximum length (123)

Case Expectation

Clean close with normal code.

Case 7.3.6

Case Description

Up

Send a close frame with close code and close reason which is too long (124) - total frame payload 126 octets

Case Expectation

Clean close with protocol error code or dropped TCP connection.

Case 7.5.1

Case Description

Up

Send a close frame with invalid UTF8 payload

Case Expectation

Clean close with protocol error or invalid utf8 code or dropped TCP.

Case 7.7.1

Case Description

Up

Send close with valid close code 1000

Case Expectation

Clean close with normal or echoed code

Case 7.7.2

Case Description

Up

Send close with valid close code 1001

Case Expectation

Clean close with normal or echoed code

Case 7.7.3

Case Description

Up

Send close with valid close code 1002

**Case Expectation**  
Clean close with normal or echoed code

Case 7.7.4

**Case Description** Up  
Send close with valid close code 1003

**Case Expectation**  
Clean close with normal or echoed code

Case 7.7.5

**Case Description** Up  
Send close with valid close code 1007

**Case Expectation**  
Clean close with normal or echoed code

Case 7.7.6

**Case Description** Up  
Send close with valid close code 1008

**Case Expectation**  
Clean close with normal or echoed code

Case 7.7.7

**Case Description** Up  
Send close with valid close code 1009

**Case Expectation**  
Clean close with normal or echoed code

Case 7.7.8

**Case Description** Up  
Send close with valid close code 1010

**Case Expectation**

Clean close with normal or echoed code

**Case 7.7.9**

**Case Description**

Up

Send close with valid close code 1011

**Case Expectation**

Clean close with normal or echoed code

**Case 7.7.10**

**Case Description**

Up

Send close with valid close code 3000

**Case Expectation**

Clean close with normal or echoed code

**Case 7.7.11**

**Case Description**

Up

Send close with valid close code 3999

**Case Expectation**

Clean close with normal or echoed code

**Case 7.7.12**

**Case Description**

Up

Send close with valid close code 4000

**Case Expectation**

Clean close with normal or echoed code

**Case 7.7.13**

**Case Description**

Up

Send close with valid close code 4999

**Case Expectation**

Clean close with normal or echoed code

## Case 7.9.1

Case Description

Up

Send close with invalid close code 0

Case Expectation

Clean close with protocol error code or drop TCP

## Case 7.9.2

Case Description

Up

Send close with invalid close code 999

Case Expectation

Clean close with protocol error code or drop TCP

## Case 7.9.3

Case Description

Up

Send close with invalid close code 1004

Case Expectation

Clean close with protocol error code or drop TCP

## Case 7.9.4

Case Description

Up

Send close with invalid close code 1005

Case Expectation

Clean close with protocol error code or drop TCP

## Case 7.9.5

Case Description

Up

Send close with invalid close code 1006

Case Expectation

Clean close with protocol error code or drop TCP

## Case 7.9.6

<b>Case Description</b>	Up
Send close with invalid close code 1012	
<b>Case Expectation</b>	
Clean close with protocol error code or drop TCP	

## Case 7.9.7

<b>Case Description</b>	Up
Send close with invalid close code 1013	
<b>Case Expectation</b>	
Clean close with protocol error code or drop TCP	

## Case 7.9.8

<b>Case Description</b>	Up
Send close with invalid close code 1014	
<b>Case Expectation</b>	
Clean close with protocol error code or drop TCP	

## Case 7.9.9

<b>Case Description</b>	Up
Send close with invalid close code 1015	
<b>Case Expectation</b>	
Clean close with protocol error code or drop TCP	

## Case 7.9.10

<b>Case Description</b>	Up
Send close with invalid close code 1016	
<b>Case Expectation</b>	
Clean close with protocol error code or drop TCP	

## Case 7.9.11

<div><div>Case Description</div><div>Send close with invalid close code 1100</div></div>	Up
<div><div>Case Expectation</div><div>Clean close with protocol error code or drop TCP</div></div>	

Case 7.9.12

<div><div>Case Description</div><div>Send close with invalid close code 2000</div></div>	Up
<div><div>Case Expectation</div><div>Clean close with protocol error code or drop TCP</div></div>	

Case 7.9.13

<div><div>Case Description</div><div>Send close with invalid close code 2999</div></div>	Up
<div><div>Case Expectation</div><div>Clean close with protocol error code or drop TCP</div></div>	

Case 7.13.1

<div><div>Case Description</div><div>Send close with close code 5000</div></div>	Up
<div><div>Case Expectation</div><div>Actual events are undefined by the spec.</div></div>	

Case 7.13.2

<div><div>Case Description</div><div>Send close with close code 65536</div></div>	Up
<div><div>Case Expectation</div><div>Actual events are undefined by the spec.</div></div>	

Case 9.1.1

<div><div>Case Description</div></div>	Up
--	----

Send text message message with payload of length  $64 * 2^{10}$  (64k).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.1.2

**Case Description** Up

Send text message message with payload of length  $256 * 2^{10}$  (256k).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.1.3

**Case Description** Up

Send text message message with payload of length  $1 * 2^{20}$  (1M).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.1.4

**Case Description** Up

Send text message message with payload of length  $4 * 2^{20}$  (4M).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.1.5

**Case Description** Up

Send text message message with payload of length  $8 * 2^{20}$  (8M).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.1.6

**Case Description** Up

Send text message message with payload of length  $16 * 2^{20}$  (16M).

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.2.1**

**Case Description**

Up

Send binary message message with payload of length  $64 * 2^{10}$  (64k).

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

**Case 9.2.2**

**Case Description**

Up

Send binary message message with payload of length  $256 * 2^{10}$  (256k).

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

**Case 9.2.3**

**Case Description**

Up

Send binary message message with payload of length  $1 * 2^{20}$  (1M).

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

**Case 9.2.4**

**Case Description**

Up

Send binary message message with payload of length  $4 * 2^{20}$  (4M).

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

**Case 9.2.5**

**Case Description**

Up

Send binary message message with payload of length  $8 * 2^{20}$  (16M).

**Case Expectation**



Receive echo'ed binary message (with payload as sent).

## Case 9.2.6

Case Description

Send binary message message with payload of length  $16 * 2^{20}$  (16M).

Up

Case Expectation

Receive echo'ed binary message (with payload as sent).

## Case 9.3.1

Case Description

Send fragmented text message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 64.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.3.2

Case Description

Send fragmented text message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 256.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.3.3

Case Description

Send fragmented text message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 1k.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.3.4

Case Description

Send fragmented text message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 4k.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

### Case 9.3.5

<b>Case Description</b>	Up
Send fragmented text message message with message payload of length 4 * 2**20 (4M). Sent out in fragments of 16k.	
<b>Case Expectation</b>	
Receive echo'ed text message (with payload as sent).	

### Case 9.3.6

<b>Case Description</b>	Up
Send fragmented text message message with message payload of length 4 * 2**20 (4M). Sent out in fragments of 64k.	
<b>Case Expectation</b>	
Receive echo'ed text message (with payload as sent).	

### Case 9.3.7

<b>Case Description</b>	Up
Send fragmented text message message with message payload of length 4 * 2**20 (4M). Sent out in fragments of 256k.	
<b>Case Expectation</b>	
Receive echo'ed text message (with payload as sent).	

### Case 9.3.8

<b>Case Description</b>	Up
Send fragmented text message message with message payload of length 4 * 2**20 (4M). Sent out in fragments of 1M.	
<b>Case Expectation</b>	
Receive echo'ed text message (with payload as sent).	

### Case 9.3.9

<b>Case Description</b>	Up
Send fragmented text message message with message payload of length 4 * 2**20 (8M). Sent out in fragments of 4M.	
<b>Case Expectation</b>	
Receive echo'ed text message (with payload as sent).	

### Case 9.4.1

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{**}20$  (4M). Sent out in fragments of 64.

Case Expectation

Receive echo'ed binary message (with payload as sent).

Case 9.4.2

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{**}20$  (4M). Sent out in fragments of 256.

Case Expectation

Receive echo'ed binary message (with payload as sent).

Case 9.4.3

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{**}20$  (4M). Sent out in fragments of 1k.

Case Expectation

Receive echo'ed binary message (with payload as sent).

Case 9.4.4

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{**}20$  (4M). Sent out in fragments of 4k.

Case Expectation

Receive echo'ed binary message (with payload as sent).

Case 9.4.5

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{**}20$  (4M). Sent out in fragments of 16k.

Case Expectation

Receive echo'ed binary message (with payload as sent).

Case 9.4.6

Up

Case Description

Send fragmented binary message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 64k.

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

## Case 9.4.7

**Case Description** Up

Send fragmented binary message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 256k.

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

## Case 9.4.8

**Case Description** Up

Send fragmented binary message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 1M.

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

## Case 9.4.9

**Case Description** Up

Send fragmented binary message message with message payload of length  $4 * 2^{20}$  (4M). Sent out in fragments of 4M.

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

## Case 9.5.1

**Case Description** Up

Send text message message with payload of length  $1 * 2^{20}$  (1M). Sent out data in chops of 64 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

## Case 9.5.2

**Case Description** Up

Send text message message with payload of length  $1 * 2^{20}$  (1M). Sent out data in chops of 128 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.5.3**

**Case Description**

Up

Send text message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 256 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.5.4**

**Case Description**

Up

Send text message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 512 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.5.5**

**Case Description**

Up

Send text message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 1024 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.5.6**

**Case Description**

Up

Send text message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 2048 octets.

**Case Expectation**

Receive echo'ed text message (with payload as sent).

**Case 9.6.1**

**Case Description**

Up

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 64 octets.

**Case Expectation**

Receive echo'ed binary message (with payload as sent).

## Case 9.6.2

Case Description

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 128 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.6.3

Case Description

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 256 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.6.4

Case Description

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 512 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.6.5

Case Description

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 1024 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.6.6

Case Description

Send binary message message with payload of length 1 \* 2\*\*20 (1M). Sent out data in chops of 2048 octets.

Up

Case Expectation

Receive echo'ed text message (with payload as sent).

## Case 9.7.1

<b>Case Description</b>	Up
Send 1000 text messages of payload size 0 to measure implementation/network RTT (round trip time) / latency.	
<b>Case Expectation</b>	
Receive echo'ed text messages (with payload as sent). Timeout case after 60 secs.	

## Case 9.7.2

<b>Case Description</b>	Up
Send 1000 text messages of payload size 16 to measure implementation/network RTT (round trip time) / latency.	
<b>Case Expectation</b>	
Receive echo'ed text messages (with payload as sent). Timeout case after 60 secs.	

## Case 9.7.3

<b>Case Description</b>	Up
Send 1000 text messages of payload size 64 to measure implementation/network RTT (round trip time) / latency.	
<b>Case Expectation</b>	
Receive echo'ed text messages (with payload as sent). Timeout case after 60 secs.	

## Case 9.7.4

<b>Case Description</b>	Up
Send 1000 text messages of payload size 256 to measure implementation/network RTT (round trip time) / latency.	
<b>Case Expectation</b>	
Receive echo'ed text messages (with payload as sent). Timeout case after 120 secs.	

## Case 9.7.5

<b>Case Description</b>	Up
Send 1000 text messages of payload size 1024 to measure implementation/network RTT (round trip time) / latency.	
<b>Case Expectation</b>	
Receive echo'ed text messages (with payload as sent). Timeout case after 240 secs.	

## Case 9.7.6

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 text messages of payload size 4096 to measure implementation/network RTT (round trip time) / latency.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed text messages (with payload as sent). Timeout case after 480 secs.</div></div>

Case 9.8.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 binary messages of payload size 0 to measure implementation/network RTT (round trip time) / latency.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed binary messages (with payload as sent). Timeout case after 60 secs.</div></div>

Case 9.8.2

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 binary messages of payload size 16 to measure implementation/network RTT (round trip time) / latency.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed binary messages (with payload as sent). Timeout case after 60 secs.</div></div>

Case 9.8.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 binary messages of payload size 64 to measure implementation/network RTT (round trip time) / latency.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed binary messages (with payload as sent). Timeout case after 60 secs.</div></div>

Case 9.8.4

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 binary messages of payload size 256 to measure implementation/network RTT (round trip time) / latency.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed binary messages (with payload as sent). Timeout case after 120 secs.</div></div>

Case 9.8.5

<div><div>Up</div><div><b>Case Description</b></div><div></div></div>
---



Send 1000 binary messages of payload size 1024 to measure implementation/network RTT (round trip time) / latency.

**Case Expectation**

Receive echo'ed binary messages (with payload as sent). Timeout case after 240 secs.

## Case 9.8.6

**Case Description**

Up

Send 1000 binary messages of payload size 4096 to measure implementation/network RTT (round trip time) / latency.

**Case Expectation**

Receive echo'ed binary messages (with payload as sent). Timeout case after 480 secs.

## Case 10.1.1

**Case Description**

Up

Send text message with payload of length 65536 auto-fragmented with **autoFragmentSize = 1300**.

**Case Expectation**

Receive echo'ed text message (with payload as sent and transmitted frame counts as expected). Clean close with normal code.

## Case 12.1.1

**Case Description**

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.1.2

**Case Description**

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.1.3

**Case Description**

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

**Case 12.1.4**

**Case Description**

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

**Case 12.1.5**

**Case Description**

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.1.6**

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.1.7**

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.1.8**

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.9

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.10

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.11

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.12

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.13

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.1.14

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.1.15

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.1.16

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.1.17

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.1.18

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.2.1

Up

Case Description

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.2.2

Up

Case Description

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.2.3

Up

Case Description

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

## Case 12.2.4

Up

Case Description

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

## Case 12.2.5

Up

Case Description

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.6

Up

Case Description

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.7

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.8

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.9

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.10

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.11

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.2.12**

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.2.13**

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.2.14**

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.2.15**

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.2.16**

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.17

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.2.18

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.3.1

**Case Description**

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.3.2

**Case Description**

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 12.3.3

**Case Description**

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.



## Case 12.3.4

### Case Description

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use default permessage-deflate offer.

### Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

## Case 12.3.5

### Case Description

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use default permessage-deflate offer.

### Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.3.6

### Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use default permessage-deflate offer.

### Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.3.7

### Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

### Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.3.8

### Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use default permessage-deflate offer.

### Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.3.9

Up

Case Description

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 12.3.10

Up

Case Description

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 12.3.11

Up

Case Description

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 12.3.12

Up

Case Description

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 12.3.13

Up

Case Description

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use default permessage-deflate offer.

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 12.3.14

Up

Case Description

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

### Case 12.3.15

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

### Case 12.3.16

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

### Case 12.3.17

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

### Case 12.3.18

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

### Case 12.4.1

**Case Description**

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

**Case 12.4.2**

**Case Description**

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

**Case 12.4.3**

**Case Description**

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

**Case 12.4.4**

**Case Description**

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

**Case 12.4.5**

**Case Description**

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.4.6**

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.7

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.8

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.9

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.10

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.11

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.4.12

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.4.13

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.4.14

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.4.15

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.4.16

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use default permessage-deflate offer.	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 12.4.17

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use default permessage-deflate offer.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>

Case 12.4.18

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use default permessage-deflate offer.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>

Case 12.5.1

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use default permessage-deflate offer.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.</div></div>

Case 12.5.2

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use default permessage-deflate offer.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.</div></div>

Case 12.5.3

<div><div>Up</div><div><b>Case Description</b></div><div>Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use default permessage-deflate offer.</div></div>
<div><div><b>Case Expectation</b></div><div>Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.</div></div>

Case 12.5.4

<div><div>Up</div><div><b>Case Description</b></div></div>
--

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

## Case 12.5.5

**Case Description**

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.6

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.7

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.8

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.9

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use default permessage-deflate offer.



**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.5.10**

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use default permesssage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.5.11**

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use default permesssage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.5.12**

**Case Description**

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use default permesssage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.5.13**

**Case Description**

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use default permesssage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

**Case 12.5.14**

**Case Description**

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use default permesssage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.15

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.16

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.17

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 12.5.18

**Case Description**

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use default permessage-deflate offer.

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.1

**Case Description**

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 13.1.2

Case Description

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 13.1.3

Case Description

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

## Case 13.1.4

Case Description

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

## Case 13.1.5

Case Description

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.6

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.7

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.8

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.9

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.10

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.1.11

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.12

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.13

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.14

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.15

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.16

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers

(requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.17

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.1.18

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.1

Case Description

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.2.2

Case Description

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.2.3

Case Description

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

Case 13.2.4

**Case Description** Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

Case 13.2.5

**Case Description** Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.6

**Case Description** Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.7

**Case Description** Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.8

<div><div>Case Description</div><div>Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]</div></div>	Up
<div><div>Case Expectation</div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>	

Case 13.2.9

<div><div>Case Description</div><div>Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]</div></div>	Up
<div><div>Case Expectation</div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>	

Case 13.2.10

<div><div>Case Description</div><div>Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]</div></div>	Up
<div><div>Case Expectation</div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>	

Case 13.2.11

<div><div>Case Description</div><div>Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]</div></div>	Up
<div><div>Case Expectation</div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>	

Case 13.2.12

<div><div>Case Description</div><div>Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]</div></div>	Up
<div><div>Case Expectation</div><div>Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.</div></div>	

Case 13.2.13



Case DescriptionUp

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.14

Case DescriptionUp

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.15

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.16

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.2.17

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.2.18

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.3.1

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.	

## Case 13.3.2

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.	

## Case 13.3.3

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.	

## Case 13.3.4

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.	

## Case 13.3.5

Case Description

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.6

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.7

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.8

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.9

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.10

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.11

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.12

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.13

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.3.14

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.3.15

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.3.16

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.3.17

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.3.18

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.1

Case Description

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers

(requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 13.4.2

**Case Description**

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 13.4.3

**Case Description**

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

## Case 13.4.4

**Case Description**

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

## Case 13.4.5

**Case Description**

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.4.6

**Case Description**

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.7

**Case Description** Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.8

**Case Description** Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.9

**Case Description** Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.10

**Case Description** Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.11

Case DescriptionUp

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.12

Case DescriptionUp

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.13

Case DescriptionUp

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.14

Case DescriptionUp

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.15

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.16



Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.17

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.4.18

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(False, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.5.1

Case DescriptionUp

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.5.2

Case DescriptionUp

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

## Case 13.5.3

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.	

## Case 13.5.4

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.	

## Case 13.5.5

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.5.6

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.5.7

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.5.8

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.9

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.10

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.11

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.12

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.13

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.14

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.15

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.16

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.5.17

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.5.18

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.1

Case Description

Up

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.6.2

Case Description

Up

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.6.3

Case Description

Up

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

Case 13.6.4

Case Description

Up

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers

(requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

Case 13.6.5

Case Description

Up

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.6

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.7

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.8

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.9

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.10

**Case Description** Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.11

**Case Description** Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.12

**Case Description** Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.13

**Case Description** Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

**Case Expectation**

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.14

Case DescriptionUp

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.15

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.16

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.17

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.6.18

Case DescriptionUp

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 15)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

Case 13.7.1



Case DescriptionUp

Send 1000 compressed messages each of payload size 16, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.7.2

Case DescriptionUp

Send 1000 compressed messages each of payload size 64, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 60 secs.

Case 13.7.3

Case DescriptionUp

Send 1000 compressed messages each of payload size 256, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 120 secs.

Case 13.7.4

Case DescriptionUp

Send 1000 compressed messages each of payload size 1024, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 240 secs.

Case 13.7.5

Case DescriptionUp

Send 1000 compressed messages each of payload size 4096, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.6

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 8192, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.7.7

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 16384, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.7.8

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 32768, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.7.9

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 65536, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.7.10

<b>Case Description</b>	Up
Send 1000 compressed messages each of payload size 131072, auto-fragment to 0 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]	
<b>Case Expectation</b>	
Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.	

## Case 13.7.11

Case Description

Up

Send 1000 compressed messages each of payload size 8192, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.12

Case Description

Up

Send 1000 compressed messages each of payload size 16384, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.13

Case Description

Up

Send 1000 compressed messages each of payload size 32768, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.14

Case Description

Up

Send 1000 compressed messages each of payload size 65536, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.15

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 256 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.16

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 1024 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.17

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 4096 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.

## Case 13.7.18

Case Description

Up

Send 1000 compressed messages each of payload size 131072, auto-fragment to 32768 octets. Use permessage-deflate client offers (requestNoContextTakeover, requestMaxWindowBits): [(True, 8), (True, 0), (False, 0)]

Case Expectation

Receive echo'ed messages (with payload as sent). Timeout case after 480 secs.