

WebSockets

Last time we saw how to establish a WebSocket connection

Today, we'll parse and send messages over the socket



https://tools.ietf.org/html/rfc6455#section-5.2

Protocols Sidenote

- Many of the protocols used in the Internet define the order and meaning of bits that are sent
 - Sender assembles the bits of a message following the protocol
 - Send the bits through the Internet
 - Receiver interprets the bits following the same protocol to extract meaning from the bits
- Protocols enable communication using only 1's and 0's

Protocols Sidenote

- TCP/IP protocol headers shown here
- how to route a packet

0 1	2	3				
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	5 6 7 8 9 0 1 2 3 4 5 6 7 8	901				
+-	-+	+_+_+_+				
Version IHL Type of Service	e Total Length					
+_						
Identification	Flags Fragment Offse	et				
+-	-+	+_+_+_+				
Time to Live Protocol	Header Checksum					
+-	-+	+_+_+_+				
Source A	Address					
+-	-+	+_+_+_+				
Destination	n Address					
+-	-+	+_+_+_+				
Options	Paddir	ng				
+-	-+	+_+_+_+				

Example Internet Datagram Header

Routers read the IP header following this protocol to know

 Endpoints follow the TCP protocol to assemble a sequence of packets and send it to the process using the given port

0		1					2								3
0 1 2 3	45678	901	234	15	67	8	90	1	23	4	5	67	8	9	0 1
+_+_+_	+_+_+_+_+	-+-+-+	_+_+-	-+_+	+-+	⊦_+	+-+	+_+	-+-	+_+	-+	-+-	+_+	+	_+_+
	Source P	ort					Des	sti	nat	ion	Ρ	ort			
+_+_+_	+_+_+_+_+	-+-+-+	_+_+-	-+-+	+-+	⊦_+	+-+	+-+	-+-	+_+	-+	-+-	+_+	+	_+_+
			Seque	ence	e Nun	nbe	er								
+_+_+_	+_+_+_+_+	-+-+-+	_+_+-	-+-+	+-+	⊦_+	+-+	+-+	-+-	+_+	-+	-+-	+_+	+	_+_+
		Ackn	owled	lgme	ent N	Jun	nber								
+_+_+_	+_+_+_+_+	-+-+-+	_+_+-	-+_+	-+-+	⊦_+	-+-+	+_+	-+-	+_+	-+	-+-	+_+	+	_+_+
Data		UA	P R S	5 F											
Offset	Reserved	R C	s s 3	2 I					Win	dow	7				
		GK	н т 1	1 N											
+-+-+	+_+_+_+_+	-+-+-+	_+_+-	-+-+	+-+	⊦_+	+-+	+-+	-+-	+_+	-+	-+-	+_+	+	-+-+
	Checksu	m			Urgent Pointer										
+_															
Options Padding															
+-															
data															
+-+-+-	+_+_+_+_+	-+-+-+	_+_+-	-+-+	+-+	⊦_+	+-+	+_+	-+-	+_+	-+	_+_	+_+	+	_+_+

TCP Header Format

- The WebSocket protocol functions the same way
- Client and server agree to follow this protocol
- Send bits in this specific order
 - We can rely on the client following this protocol



 F | R | R | R | opcode | M | Payload len |
 Extended payload length

 I | S | S | S | (4) | A | (7) |
 (16/64)

 N | V | V | V |
 S |
 (16/64)

 1 | 2 | 3 |
 K |
 (16/10)

 Extended payload length continued, if payload len == 127 |Masking-key, if MASK set to 1 Payload Data Payload Data continued ... Payload Data continued ...

Network Stack

 An IP packet containing a WebSocket frame looks like this

0 0 1 2 3 4 5 6 7 8 9	1 9 0 1 2 3 4 5	2 3 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1						
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-	pe of Service	-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+						
Identifica	+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-							
Time to Live	Protocol	-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+						
Source Address								
	Destination							
 +_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_	Options	-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+						
Source P	ort	Destination Port						
+_	+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-							
+-	Acknowledgme	ent Number						
<pre>-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+</pre>	-+-+-+-+-+-+-+-+- U A P R S F R C S S Y I G K H T N N	Window						
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-	_+_+_+_+_+_+_+_+_ m	Urgent Pointer						
+-	-+-+-+-+-+-+-+-+ Options _+_+_+_+_+_+_+_+	-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+						
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-	Payload len (7)	++ Extended payload length (16/64) (if payload len==126/127) + +						
Extended payload length continued, if payload len == 127								
Masking-key, if MASK set to 1								
Masking-key (continued) Payload Data +								
Payload Data continued + +								
 +	Payload Da	ta continued +						

IP

TCP

WebSocket

Parsing Bits

- We will have to read frames at the bit level
 - It's already in a byte array when we receive it
 - We can access any byte and extract the bits we need
 - Helpful to recall that bytes are represented as 8-bit integer values (0-255)

0 1 2 3 4 5 6 7 8 9 0 1 2 3 +-+-+-+-+-+						_			
0 1 2 3 4 5 6 7 8 9 0 1 2 3 +-+-+++-++-+-+	0					-	1		
<pre>+-+-++++++++</pre>	012	234	5	67	8	9 (01	2	3
F R R R opcode M Payload I S S S (4) A (7) N V V V S 1 2 3 K +-+++++++++++++++++++++++++++++++++++	+_+_+-	-+-+-			+_+				
I S S S (4) A (7) N V V V S 1 2 3 K +-+-+++++++++++++++++++++++++++++++++	FRF	λ R	opc	ode	M	Pa	ayl	oad	1
N V V V S 1 2 3 K +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-	ISS	sisi	(4)	A			(7))
1 2 3 K +-+-+-+-+	N V V	<i>i</i> vi			İsİ			. ,	
+-+-+-+-++-+	1 2	2 3 1			İκİ				
<pre>Extended payload lengt +</pre>	+-+-+-	-+-+-			╊╼╋				
<pre>+</pre>	1	Exte	nde	d pa	avl	oad	d 10	enc	rt.
 Masking-key (continued) + i Payloa +	+			~					
Masking-key (continued) +	1								
Masking-key (continued) +									
Masking-key (Continued) + Payloa +			1			 :		、 	
+ Payloa +	Masi	cing-	кеу	(00	ont	int	lea)	
: Payloa +	+								
+ Payloa	:						Pa	у⊥с	ba
Payloa +	+					-			
+							Pa	ylc	ba
	+								



Parsing Bits

• Bit Example - To read the opcode:

- get the byte at index 0

- We now have an int from 0-15 representing the opcode

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+	2 3 6789012345678901						
+-+-+-+-++-+-+++++++-+							
+ +							
Masking-key (continued) Payload Data							
Payload Data continued							
+							

• Bitwise AND (& in most languages) this byte with a "bit mask" of 15 • Since 15 = 00001111 as a byte this will 0 out the 4 higher order bits

• FIN: The finish bit

- 1 This is the last frame for this message
- data for the same message

0 0	1	2	3	4	5	6	7	8	9	1 0	1	2	3
+_+ F I N	-+ R S V 1	R S V 2	R S V 3	 (opo (4	 coc 4)	de	M A S K	 I	?ay	/	 oac (7)	 1 [)
+_+ + _ +	+ -	++ ₽ 	+-+ 5xt 		nde	ed	ра 	+ ay] 		ad 	1e 	eng 	gt]
м +	as 	ski	ing	g_}	cey	Y	(co	ont	:ir	nue	ed))	
:										I	Pay	<i>y</i> lo	ba
+										I	Pay	/lo	- oa

• 0 - There will be continuation frames containing more

len | Extended payload length (16/64) (if payload len==126/127) h continued, if payload len == 127 +-----|Masking-key, if MASK set to 1 ____+ Payload Data d Data continued ... d Data continued ...

• RSV: Reserved bits Used to specify any extensions being used • [You can assume these are always 000 for the HW]

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 + + + + + + + + + + + + + + + + + + +	2 3 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1							
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	i len Extended payload length (16/64) (if payload len==126/127)							
Extended payload len	+-+-+-+-++-+-++							
	+ + + +							
Masking-key (continued) Payload Data								
Payload Data continued								
+								

opcode: Operation code

- connection, 0000 for continuation frame

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 + + + + + + + + + + + + + + + + + + +	2 3 6789012345678901						
F R R R opcode M Payload len I S S S (4) A (7) N V V V S 1 2 3 K +-+-+-+-+	Extended payload length (16/64) (if payload len==126/127) + +						
Extended payload length co	ntinued, if payload len == 127						
	Masking-key, if MASK set to 1						
+Payload Data							
Payload Data continued							
+ + Payload Data continued ++							

Specifies the type of information contained in the payload

• Ex: 0001 for text, 0010 for binary, 1000 to close the

• MASK: Mask bit

• Set to 1 if a mask is being used

• Set to 0 if no mask is being used

• This will be 1 when receiving messages from a client

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+	2 3 6789012345678901 +					
F R R opcode M Payload len Extended payload length I S S S (4) A (7) (16/64) N V V V S (if payload len==126/127) 1 2 3 K Extended payload length continued, if payload len == 127						
+ +						
Masking-key (continued) Payload Data						
Payload Data continued						
Payload Data continued						

The next bits will represent payload length in bytes
Similar to Content-Length
The length can be represented in 7, 16, or 64 bits

0 1 2 3	3456	789	1 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-	R opcod 5 (4) 7 8	-+-+ A S K	ayload (7)
E2 +	tended	payloa 	d lengt
Maskir	ng-key (contin	ued)
:			Payloa
+ 			Payloa

2 3 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 len Extended payload length (16/64) (if payload len==126/127) + - - + - - - - - - + + h continued, if payload len == 127 | Masking-key, if MASK set to 1 | Payload Data | Payload Data | Ad Data continued ...

• If the length is <126 bytes

the MASK bit

• The next bit after the length is either the mask or payload

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_+_+	2 6789012345678901 +					
+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+						
	Masking-key, if MASK set to 1					
Masking-key (continued) Payload Data						
Payload Data continued						
+						

• The length is represented in 7 bits, sharing a byte with

If the length is >=126 and <65536 bytes
The 7 bit length will be exactly 126 (1111110)
The next 16 bits represents the payload length

0 0 1 2 3 +_+_+_+	456	789	1 0 1 2 3
F R R R I S S S N V V V 1 2 3	opcod (4) 	de M A S K	Payload (7)
+_+_+_ Ex + +	tended 	paylo	ad lengt
Maskin	g-key	(conti	nued)
:			Payloa
+			Payloa
+			

- If the length is >=65536 bytes
 - The 7 bit length will be exactly 127 (1111111)
 - The next 64 bits represents the payload length
 - 18,446,744,073,709,551,615 max length!
 - 16 exabytes / 16,000,000 terabytes

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_+	2 3 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 ++	
F R R R opcode M Payload len Extended payload length I S S S (4) A (7) (16/64) N V V V S (if payload len==126/127) 1 2 3 K +-+-+-+-+-++ - Futorded payload length continued if payload len == 127		
	<pre>// Hasking-key, if MASK set to 1</pre>	
Masking-key (continued) Payload Data		
Payload Data continued		
+		

- To read the frame length, read the 7 bit length

 - Else, the value itself is the length

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_++	2 3 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 ++		
F R R R opcode M Payload len Extended payload length I S S S (4) A (7) (16/64) N V V V S (if payload len==126/127) 1 2 3 K +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-			
+ +			
Masking-key (continued) Payload Data			
Payload Data continued			
+			

• If the value is 126, read the next 16 bits as the length • If the value is 127, read the next 64 bits as the length

• After all the length bits: mask

• If the MASK bit == 0, the payload begins

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 + + + + + + + + + + + + + + + + + + +	2 3 6789012345678901		
F R R R opcode M Payload len I S S S (4) A (7) N V V V S 1 2 3 K	Extended payload length (16/64) (if payload len==126/127)		
Extended payload length continued, if payload len == 127			
	Masking-key, if MASK set to 1		
Masking-key (continued) Payload Data			
Payload Data continued			
+			

• If the MASK bit == 1, the next 4 bytes (32 bits) is the

- If there is a mask, read these 4 bytes
- message

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 + + + + + + + + + + + + + + + + + + +	2 3 6789012345678901	
+-+++++++++++++++++++++++++++++++++++		
	Masking-key, if MASK set to 1	
Masking-key (continued) Payload Data		
Payload Data continued		
+ + Payload Data continued ++		

• The mask will be randomly generated by the client for each

• You must parse this each time a message is received

- Read the payload 4 bytes at a time and XOR the bytes with the mask
- needed

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_+_+	2 3 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 ++	
F R R opcode M Payload len Extended payload length I S S S (4) A (7) (16/64) N V V V S (if payload len==126/127) 1 2 3 K +-+++++++++++++++++++++++++++++++++++		
	Masking-key, if MASK set to 1 +	
Masking-key (continued) Payload Data		
Payload Data continued		
+		

• Each 4 bytes of the payload has been XORed with the mask by the client

• If the length is not a multiple of 4, use only the bytes of the mask that are

• Ie. Always reading 4 bytes will cause an index out of bounds error

XOR Example

- If 4 bytes of the message are:
 - 01001001_01000011_01010101_00100001
- And the random mask is:
 - 01111011_00100010_01110101_01110011
- This part of the payload will be "message XOR mask":
 - 00110010_01100001_00100000_01010010
- When we receive these bits and XOR it with the mask again we get the original message bits:
 - 01001001_01000011_01010101

Once the payload is XORed with the mask 4 bytes at time we get the entire message
Then process the message

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+	2 3 6789012345678901		
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Extended payload length (16/64) (if payload len==126/127)		
Extended payload length continued, if payload len == 127			
	Masking-key, if MASK set to 1		
Masking-key (continued) Payload Data			
Payload Data continued			
+			

• To send a message to a client:

- Use this same format
- Assemble a byte array with the appropriate values

Append your payload as bytes

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_+_++	2 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +	
<pre>+-+-+-+-++-++++</pre>		
+	++ Masking-key, if MASK set to 1	
Masking-key (continued) Payload Data		
Payload Data continued		
Payload Data continued		

Do not use a mask when sending frames to a client • No caching concerns on server to client frames

0 0 1 2 3 +-+-+-+-	4567 +	1 8 9 0 1 2 3 +_+
F R R R I S S S N V V V 1 2 3	opcode (4) 	M Payload A (7) S K
Ex +	tended p	ayload lengt
Maskin	g-key (c	ontinued)
:		Payloa
		Payloa
T		



len | Extended payload length (16/64) (if payload len==126/127) h continued, if payload len == 127 |Masking-key, if MASK set to 1 ____+ Payload Data d Data continued ... d Data continued ...

• Example: For our purposes in the HW

RSVs are always 0

• opcode is either 0001 (Sending text), 1000 (close connection), or 0000 (continuation frame)

0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +_+_+_+_+	2 3 6789012345678901		
<pre>+-+-+-+-++-++</pre>			
+ +			
+Payload Data			
Payload Data continued			
Payload Data continued			

bits are needed for the length

messages

0			1
0 1 2 3	456	789	0 1 2 3
+-+-+-+	+	_+_+	
F R R R	opcod	e M P	ayload
I S S S	(4)	A	(7)
N V V V		S	
1 2 3		K	
+-+-+-	+	_+_+	
Ex Ex	tended	payloa	d lengt
+			
+			
Maskin	g-key (contin	ued)
+			
:			Payloa
+			
			Payloa
+			

- Check the length of your payload to determine how many
- Follow the same format for payload length as the received



MASK bit is 0 and there are not mask bytes After payload length, immediately add the bytes of the payload

0 0 1 2 3 4 5 6 7 8 9 +-+-+-+	1 0 1 2 3
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	ayload (7)
Extended payloa +	d lengtl
Masking-key (conting)	ued)
•	Payload
+	Payload

2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 len Extended payload length (16/64) (if payload len==126/127) h continued, if payload len == 127 Masking-key, if MASK set to 1 Payload Data d Data continued ...