

Part1

Browser.png



question a

Wireshark packet capture showing an HTTP GET request for /~guidot/image.php?id=szha414. The packet list shows packet 47 as the selected packet. The packet details pane shows the full request line and headers. The packet bytes pane shows the raw data in hexadecimal and ASCII.

Filter: `dst == 130.194.66.43 && ip.addr == 43.246.235.99 || (ip.dst == 43.246.235.99 && ip.addr == 130.194.66.43) || http contains "http://s-cah-mwallace.infotech.monash.edu/~guidot/image.php?id=szha414"`

No.	Time	Source	Destination	Protocol	Length	Info
42	1.022187	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=913427872 TSecr=1127123379
43	1.022305	130.194.66.43	43.246.235.99	TCP	74	80 → 51315 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1127123379 TSecr=9...
44	1.022307	130.194.66.43	43.246.235.99	TCP	74	80 → 51313 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1127123379 TSecr=913427...
45	1.022332	43.246.235.99	130.194.66.43	TCP	66	51315 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=913427872 TSecr=1127123379
46	1.022377	43.246.235.99	130.194.66.43	TCP	66	51313 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=913427872 TSecr=1127123379
47	1.032100	43.246.235.99	130.194.66.43	HTTP	962	GET /~guidot/image.php?id=szha414 HTTP/1.1
48	1.034630	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [ACK] Seq=1 Ack=897 Win=31872 Len=0 TSval=1127123382 TSecr=913427881
49	1.035946	130.194.66.43	43.246.235.99	HTTP	787	HTTP/1.1 200 OK (text/html)
50	1.035988	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=897 Ack=722 Win=131040 Len=0 TSval=913427884 TSecr=1127123382
52	1.178289	43.246.235.99	130.194.66.43	HTTP	900	GET /~guidot/image_internal.php?id=szha414 HTTP/1.1
53	1.212166	43.246.235.99	130.194.66.43	TCP	900	[TCP Retransmission] 51314 → 80 [PSH, ACK] Seq=897 Ack=722 Win=131072 Len=834 TSval=913428059 TSecr=...
54	1.214072	130.194.66.43	43.246.235.99	TCP	78	80 → 51314 [ACK] Seq=722 Ack=1731 Win=34816 Len=0 TSval=1127123427 TSecr=913428059 SLE=897 SRE=1731

Frame 47: 962 bytes on wire (7696 bits), 962 bytes captured (7696 bits) on interface 0
Ethernet II, Src: Apple_8e:2f:f4 (98:Sa:eb:8e:2f:f4), Dst: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)
Internet Protocol Version 4, Src: 43.246.235.99, Dst: 130.194.66.43
Transmission Control Protocol, Src Port: 51314 (51314), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 896
Hypertext Transfer Protocol
GET /~guidot/image.php?id=szha414 HTTP/1.1\r\nHost: s-cah-mwallace.infotech.monash.edu\r\nConnection: keep-alive\r\nCache-Control: max-age=0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36\r\nAccept-Encoding: gzip, deflate, sdch\r\nAccept-Language: zh-CN,zh;q=0.8,en;q=0.6,zh-TW;q=0.4\r\nCookie: [truncated]Cookie: SQ_SYSTEM_SESSION=015dqrblva7af4jn17ak61e6uqcl2j2011induoj2a6303ir7bjeln47mop16ltnhavdb02r89vd8ptfbfspbkh34mlnsj741s69g2; __utma=259431062.1555507830.1441034\r\nIf-Modified-Since: Sun, 10 Apr 2016 03:41:52 GMT\r\n\r\n[Full request URI: http://s-cah-mwallace.infotech.monash.edu/~guidot/image.php?id=szha414]
[HTTP request 1/2]
[Response in frame: 49]
[Next request in frame: 52]

0040 10001001 10110011 01000111 01000101 01010100 00100000 00101111 01111110 ..GET /\r\n0048 01100111 01110101 01101001 01100100 01101111 01110100 00101111 01101001 guidot/i\r\n0050 01101101 01100001 01100111 01100101 00101110 01110000 01101000 01101000 mage.php\r\n0058 00111111 01101001 01100100 00111101 01110011 01111010 01101000 01100001 id=szha\r\n0060 00110100 00110001 00110100 00100000 01001000 01010100 01010100 01010000 414 HTTP\r\n0068 00101111 00110001 00101110 00110001 00001101 00001010 01001000 01101111 /1.1..Ho\r\n0070 01110011 01110100 00110100 00100000 01110011 00101101 01100011 01100001 st: s-ca\r\n0078 01101000 00101101 01101101 01101111 01100001 01101100 01101100 01100001 h-mwalla

- packet number: 47
- request line: "GET /~guidot/image.php?id=szha414 HTTP/1.1\r\n"

question b

47	1.032100	43.246.235.99	130.194.66.43	HTTP	962	GET /~guidot/image.php?id=szha414 HTTP/1.1
48	1.034630	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [ACK] Seq=1 Ack=897 Win=31872 Len=0 TSval=1127123382 TSecr=913427881
49	1.035946	130.194.66.43	43.246.235.99	HTTP	787	HTTP/1.1 200 OK (text/html)
50	1.035988	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=897 Ack=722 Win=131040 Len=0 TSval=913427884 TSecr=1127123382
52	1.178289	43.246.235.99	130.194.66.43	HTTP	908	GET /~guidot/image_internal.php?id=szha414 HTTP/1.1
▶ Frame 47: 962 bytes on wire (7696 bits), 962 bytes captured (7696 bits) on interface 0						
▼ Ethernet II, Src: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4), Dst: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)						
▼ Destination: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)						
Address: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)						
.... ..0. = LG bit: Globally unique address (factory default)						
.... ..0. = IG bit: Individual address (unicast)						
▼ Source: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4)						
Address: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4)						
.... ..0. = LG bit: Globally unique address (factory default)						
.... ..0. = IG bit: Individual address (unicast)						
Type: IPv4 (0x0800)						
▶ Internet Protocol Version 4, Src: 43.246.235.99, Dst: 130.194.66.43						
▶ Transmission Control Protocol, Src Port: 51314 (51314), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 896						

- source device: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4)
- destination device: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)

question c

method1

52	1.178289	43.246.235.99	130.194.66.43	HTTP	900	GET /~guidot/image_internal.php?id=szha414 HTTP/1.1
53	1.212166	43.246.235.99	130.194.66.43	TCP	900	[TCP Retransmission] 51314 → 80 [PSH, ACK] Seq=897 Ack=722 Win=131072 Len=834 TSval=913428...
54	1.214072	130.194.66.43	43.246.235.99	TCP	78	80 → 51314 [ACK] Seq=722 Ack=1731 Win=34816 Len=0 TSval=1127123427 TSecr=913428059 SLE=897...
55	1.385909	130.194.66.43	43.246.235.99	TCP	1514	[TCP segment of a reassembled PDU]
56	1.385912	130.194.66.43	43.246.235.99	TCP	1514	[TCP segment of a reassembled PDU]
57	1.385913	130.194.66.43	43.246.235.99	TCP	1514	[TCP segment of a reassembled PDU]
58	1.385914	130.194.66.43	43.246.235.99	TCP	1514	[TCP segment of a reassembled PDU]

Frame 55: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

Ethernet II, Src: CiscoInc_58:6d:c0 (c8:f9:f9:58:6d:c0), Dst: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4)

Internet Protocol Version 4, Src: 130.194.66.43, Dst: 43.246.235.99

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 51314 (51314), Seq: 722, Ack: 1731, Len: 1448

Source Port: 80

Destination Port: 51314

[Stream index: 4]

[TCP Segment Len: 1448]

Sequence number: 722 (relative sequence number)

[Next sequence number: 2170 (relative sequence number)]

Acknowledgment number: 1731 (relative ack number)

Header Length: 32 bytes

Flags: 0x010 (ACK)

Window size value: 272

[Calculated window size: 34816]

[Window size scaling factor: 128]

Checksum: 0xd83f [validation disabled]

Urgent pointer: 0

Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

[SEQ/ACK analysis]

TCP segment data (1448 bytes)

Packet 55 is the first image data packet, and there are hundreds of similar packet, so we can estimate the efficiency by computing the efficiency of this packet.

According to the picture, the packet length is 1514 bytes, and the length of this segment is 1448 bytes, so the answer is 0.956 (1448/1514)

method2

697	1.471891	130.194.66.43	43.246.235.99	TCP	1514 [TCP Out-Of-Order] 80 → 51314 [PSH, ACK] Seq=55...
698	1.471922	43.246.235.99	130.194.66.43	TCP	66 51314 → 80 [ACK] Seq=1731 Ack=553507 Win=128512...
[Frame: 651, payload: 518384-519831 (1448 bytes)]					
[Frame: 652, payload: 519832-521279 (1448 bytes)]					
[Frame: 656, payload: 521280-522727 (1448 bytes)]					
[Frame: 658, payload: 522728-524175 (1448 bytes)]					
[Frame: 659, payload: 524176-525623 (1448 bytes)]					
[Frame: 660, payload: 525624-527071 (1448 bytes)]					
[Frame: 661, payload: 527072-528519 (1448 bytes)]					
[Frame: 662, payload: 528520-529967 (1448 bytes)]					
[Frame: 663, payload: 529968-531415 (1448 bytes)]					
[Frame: 664, payload: 531416-532863 (1448 bytes)]					
[Frame: 671, payload: 532864-534311 (1448 bytes)]					
[Frame: 672, payload: 534312-535759 (1448 bytes)]					
[Frame: 673, payload: 535760-537207 (1448 bytes)]					
[Frame: 675, payload: 537208-538655 (1448 bytes)]					
[Frame: 676, payload: 538656-540103 (1448 bytes)]					
[Frame: 677, payload: 540104-541551 (1448 bytes)]					
[Frame: 678, payload: 541552-542999 (1448 bytes)]					
[Frame: 679, payload: 543000-544447 (1448 bytes)]					
[Frame: 680, payload: 544448-545895 (1448 bytes)]					
[Frame: 681, payload: 545896-547343 (1448 bytes)]					
[Frame: 682, payload: 547344-548791 (1448 bytes)]					
[Frame: 683, payload: 548792-550239 (1448 bytes)]					
[Frame: 697, payload: 550240-551687 (1448 bytes)]					
[Frame: 674, payload: 551688-552784 (1097 bytes)]					
[Segment count: 382]					
[Reassembled TCP length: 552785]					
[Reassembled TCP Data: 485454502f312e312032303204f4b0d0a446174653a2053...]					

There are 382 packets, each has 1514 bytes length, and the total length of the data is 552784 bytes, so the efficiency is $552784 / (1514 * 382) = 0.955$.

question d

- keep-alive: the connection can be reused in a time range.
- close: close the connection once the client has received all data.
- advantage:

Take request a web page as an example. The client request for not only the web page itself, but also other dependent documents. By Using keep-alive, only one TCP connection need to be seted up, otherwise it will set up a TCP connection for every dependent documents, which is unefficient.

What's more, by using keep-alive, it is possible that let server push some information to client forwardly, so the web page can display real time information without refresh frequently.

question e

697	1.471891	130.194.66.43	43.246.235.99	TCP	1514	[TCP Out-Of-Order] 80 → 51314 [PSH, ACK] Seq=55...
698	1.471922	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=1731 Ack=553507 Win=128512...
725	6.499619	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [FIN, ACK] Seq=553507 Ack=1731 Win=3...
726	6.499676	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=1731 Ack=553508 Win=131072...
732	12.584688	43.246.235.99	130.194.66.43	TCP	66	51315 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131744 Le...
733	12.585335	43.246.235.99	130.194.66.43	TCP	66	51313 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131744 Le...
734	12.585335	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [FIN, ACK] Seq=1731 Ack=553508 Win=1...
736	12.587070	130.194.66.43	43.246.235.99	TCP	66	80 → 51315 [FIN, ACK] Seq=1 Ack=2 Win=29056 Len...
737	12.587145	43.246.235.99	130.194.66.43	TCP	66	51315 → 80 [ACK] Seq=2 Ack=2 Win=131744 Len=0 T...
738	12.588476	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [ACK] Seq=553508 Ack=1732 Win=34816 ...
739	12.588479	130.194.66.43	43.246.235.99	TCP	66	80 → 51313 [FIN, ACK] Seq=1 Ack=2 Win=29056 Len...
740	12.588574	43.246.235.99	130.194.66.43	TCP	66	51313 → 80 [ACK] Seq=2 Ack=2 Win=131744 Len=0 T...

Packet 725: the server is going to close, so it sends a `FIN` to client(port:51314) and change it's status to `FIN wait 1`. (The `ACK` is for the previous packet 698)

Packet 734: the client received the `FIN`. It replies the `ACK` to this `FIN`, and sends a `FIN` to server to indicates that it's going to close as well. Now the client's status is `FIN wait 1`.

Packet 738: the server received the `ACK` for the previous `FIN`, it can be closed now, and it replies a `ACK` for client's `FIN`. The client receives the `ACK` from server, and it can be closed now.

question f

46	1.022377	43.246.235.99	130.194.66.43	TCP	66	51313 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 T...
47	1.032100	43.246.235.99	130.194.66.43	HTTP	962	GET /~guidot/image.php?id=szha414 HTTP/1.1
48	1.034630	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [ACK] Seq=1 Ack=897 Win=31872 Len=0 ...
Frame 47: 962 bytes on wire (7696 bits), 962 bytes captured (7696 bits) on interface 0						
Ethernet II, Src: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4), Dst: CiscoInc_9f:f0:04 (00:00:0c:9f:f0:04)						
Internet Protocol Version 4, Src: 43.246.235.99, Dst: 130.194.66.43						
Transmission Control Protocol, Src Port: 51314 (51314), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 896						
Source Port: 51314						
Destination Port: 80						
48	1.034630	130.194.66.43	43.246.235.99	TCP	66	80 → 51314 [ACK] Seq=1 Ack=897 Win=31872 Len=0 ...
49	1.035946	130.194.66.43	43.246.235.99	HTTP	787	HTTP/1.1 200 OK (text/html)
50	1.035988	43.246.235.99	130.194.66.43	TCP	66	51314 → 80 [ACK] Seq=897 Ack=722 Win=131040 Len...
51	1.130300	43.246.235.99	130.194.66.43	HTTP	962	GET /~guidot/image.php?id=szha414 HTTP/1.1
Frame 49: 787 bytes on wire (6296 bits), 787 bytes captured (6296 bits) on interface 0						
Ethernet II, Src: CiscoInc_58:6d:c0 (c8:f9:f9:58:6d:c0), Dst: Apple_8e:2f:f4 (98:5a:eb:8e:2f:f4)						
Internet Protocol Version 4, Src: 130.194.66.43, Dst: 43.246.235.99						
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 51314 (51314), Seq: 1, Ack: 897, Len: 721						
Source Port: 80						
Destination Port: 51314						

Packet 47 client -> server (seq=1, ack=1, len=896):

Client sends a http request to sever. seq=1 means this is thnn sequence number for the 1st byte in the segment, ack=1 means the client expect the server send the by te start from the sequence number = 1.

Packet 48 server -> client (seq=1, ack=897):

Server send a ACK for previous request. seq=1 because this is the sequence number for the 1st byte in the segment as well, ack=897, because the server has received 896 bytes data from client, and it expects the byte start from the sequence number = 897.

Packet 49 server -> client (seq=1, ack=897, len=721)

After replies ACK, server begin to send data.

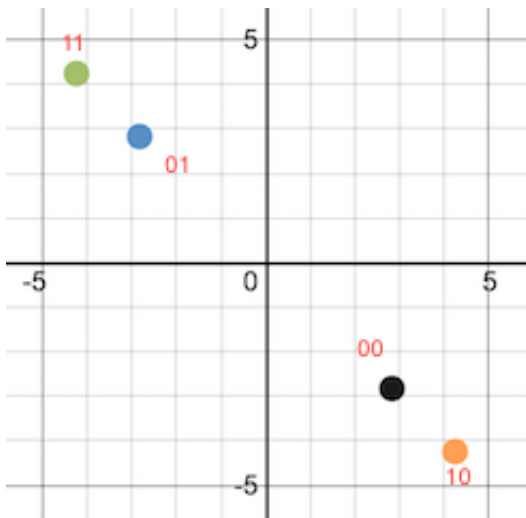
Packet 50 client -> server (seq=897, ack=722)

The client has received the data from server and replies a ack. seq=897 because th is message start from sequence number 897, ack=722 because it has received 721 byt es, and it expect the byte start from the sequence number = 722.

Part2

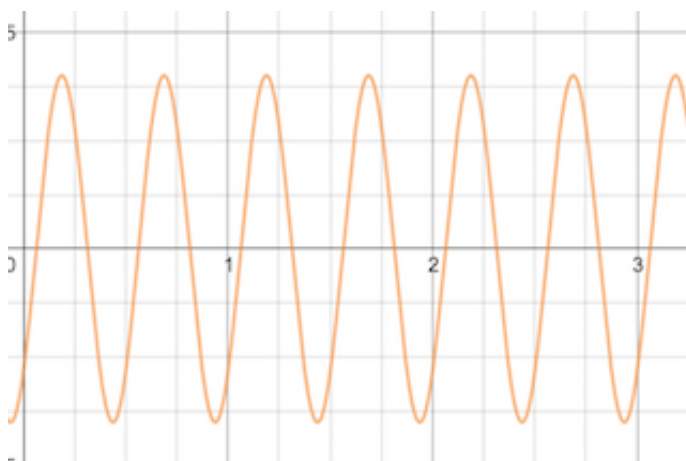
Modulation A

- type of coding: (AM, PM)
- size of symbol: 2 bits
- constellation diagram

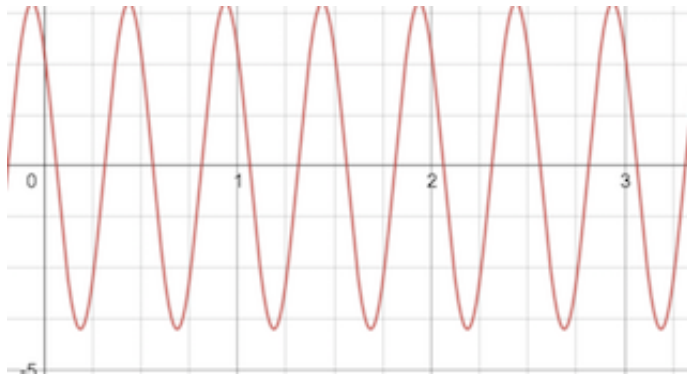


- link: <https://www.desmos.com/calculator/nb25yhooet>

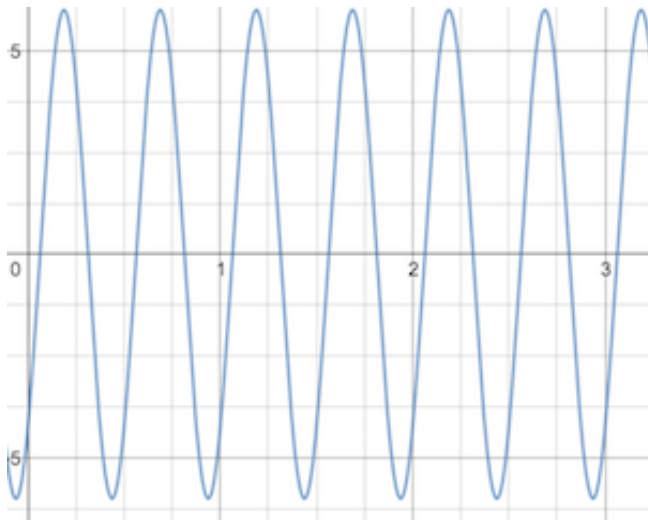
- wave form



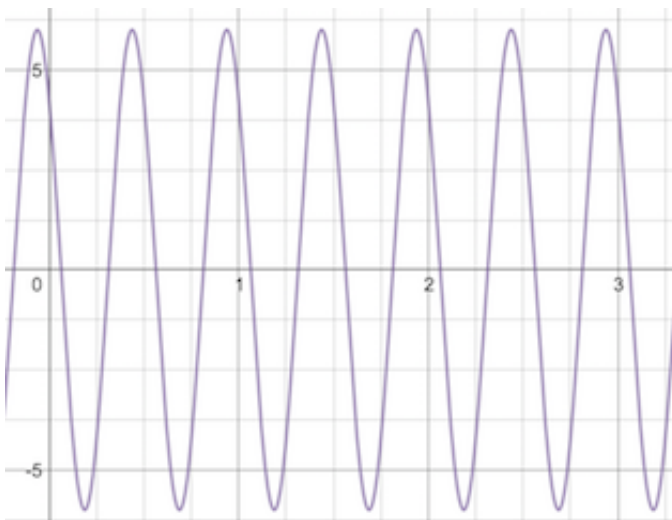
- 00 =



◦ **01**



◦ **10**



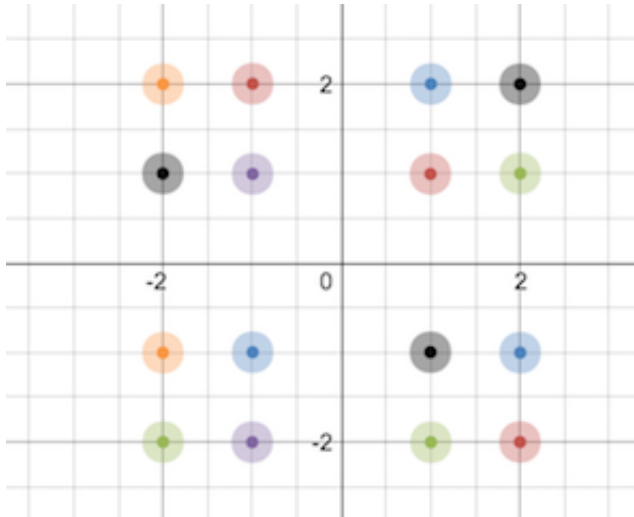
◦ **11**

◦ **link:** <https://www.desmos.com/calculator/avf1jtlkix>

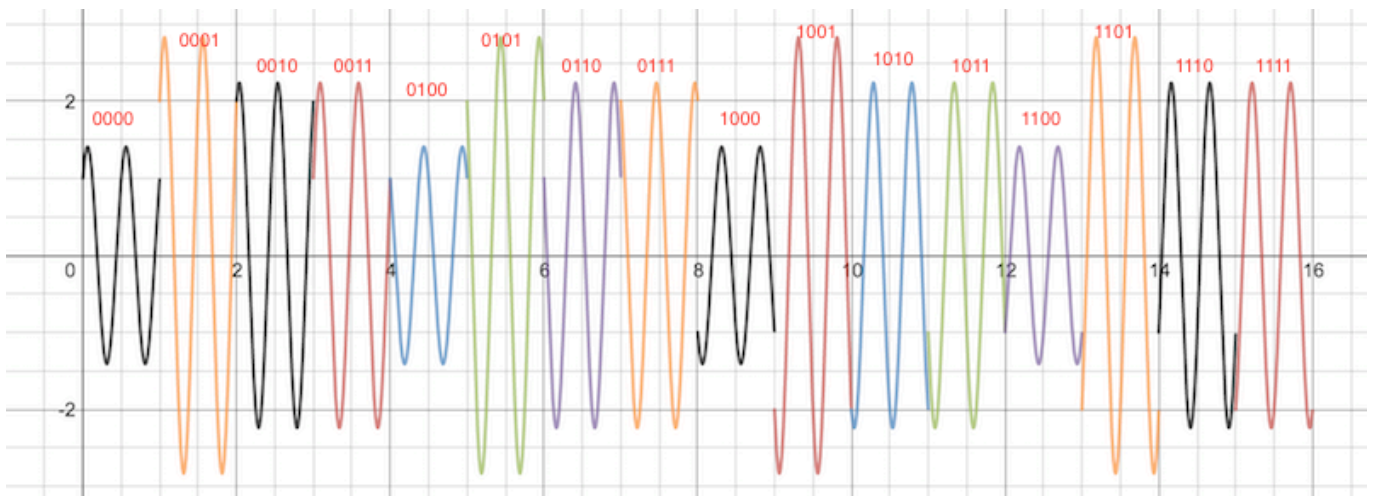
• demodulate: 0010110011010010

Modulation B

- type of coding: (AM, PM)
- size of symbol: 4 bits
- constellation diagram



- wave form

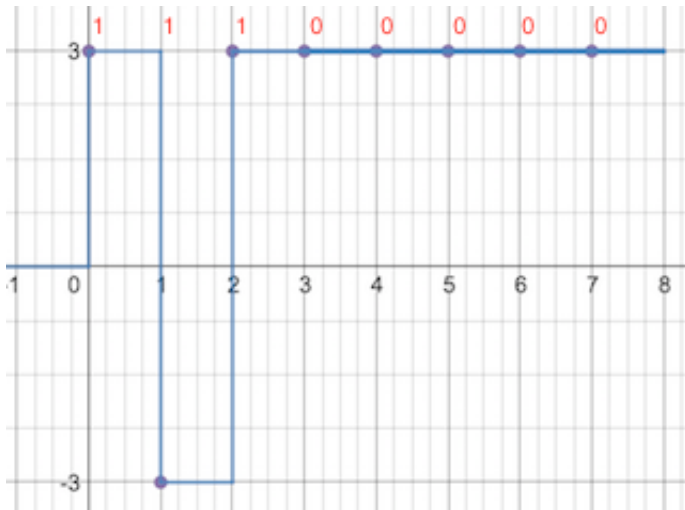


- link: <https://www.desmos.com/calculator/6hz2qzxfn6>

Digital encoding

Bipolar Non Return to Zero Inverted

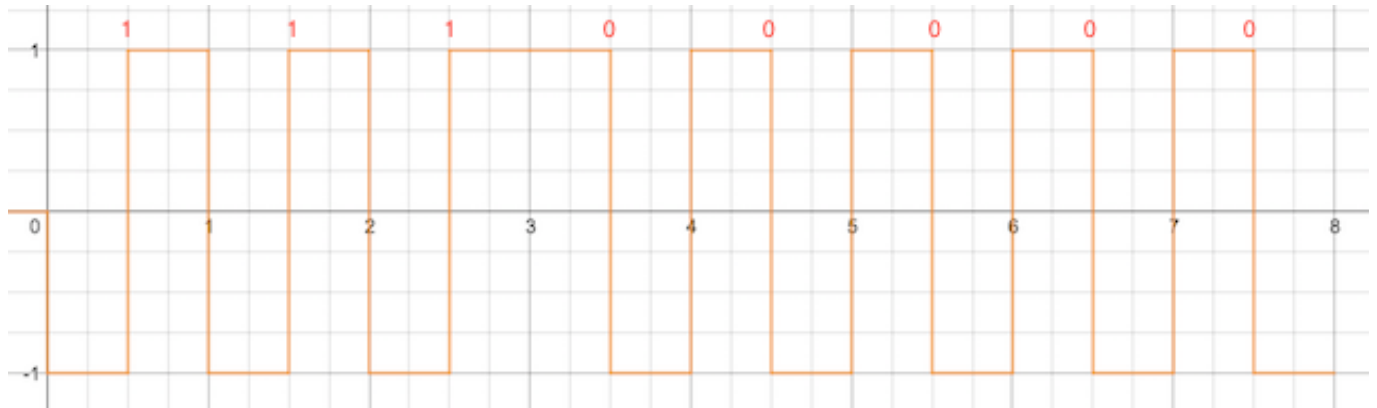
- 0: signal does not have transmitted.
- 1: signal has transmitted at clock boundary.
- wave form



- link: <https://www.desmos.com/calculator/tidpahuycx>

Manchester Encoding

- 0: low to high in the middle
- 1: high to low in the middle
- wave form



- link: <https://www.desmos.com/calculator/fonqchgy7s>