Brandon Vo

Source Engine Multiplayer Matchmaking

Online multiplayer presents a unique challenge for game developers regarding how to select and gather groups of players for a session. Normally, players don't put much thought into what server to join because the process of searching for an active server has been streamlined through the game itself through matchmaking. Many games featuring online matchmaking solve the problem by simply directing the player to the nearest server relative to his/her location akin to anycast. However, this challenge becomes more complicated for online games which need to take user preference into account. The company known as Valve Software has incorporated user preference as a part of their own matchmaking system into their game engine, the Source Engine, where users would be presented with a list of various servers and decide which one to join. Valve's networking system was created using a modified set of TCP and UDP protocols such that it can allow any Source Engine games to easily establish an online multiplayer network and advertise themselves without any prior set up needed.

Steam's Network Architecture

Because Valve Software's platform serves as both a gaming platform and a hosting service for their games, the company also acts as the main platform which bridges the connection between a client and a game's servers. The goal of Source Engine's networking is to allow anybody to create and host a server which can be advertised by Valve's platform, Steam. To achieve this, they use an architectural infrastructure similar to the Master/Slave architecture used in DNS and other multiplayer matchmaking platforms. When a server wants to be advertised, it will connect to the master server using a set of modified UDP packets using Steam's headers and takes place on port 27015[1].

The master server is responsible for managing and keeping track of all servers of a specified game similar to how DNS keeps track of the relationships between IP Addresses and domain names[2]. These master servers are owned by and managed by Valve, giving them access to player and server statistics, and the master server keeps track of all available servers, the IP Addresses being used, and the port number needed to connect to them. Any players who want to find a game will first connect to the master server and request a query of all available servers found.

Obtaining a query of all available servers

When a player starts a game, the first thing he/she must do is find a server to play in. The user client will send a query request to what is known as the master server, a host that is responsible for maintaining a list of all known servers similar to a root DNS server. As such, clients would send a UDP Request message which can be seen as an abbreviated and modified version of a DNS request packet for a list of servers. This process is known as the Master Server Query Protocol[3]. For these messages, the UDP header will contain FF FF FF FF 54 53 to identify its packet as the Source Engine Query protocol[4].

An example of the packet would be the hex packet shown below. This is an old version of the Master Server Query Protocol which uses a slightly different header label for a request message[5].

[1] https://help.steampowered.com/en/faqs/view/2EA8-4D75-DA21-31EB
[2] https://docs.linuxgsm.com/steamcmd/steam-master-server
[3] https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol
[4] https://steamcommunity.com/discussions/forum/14/2989789048633291344/
[5] https://steamcommunity.com/discussions/forum/14/2989789048633291344/

```
31 ff 30 2e 30 2e 30 2e 30 3a 30 00 5c 6e 61 70 70 5c 35 30 30 00
```

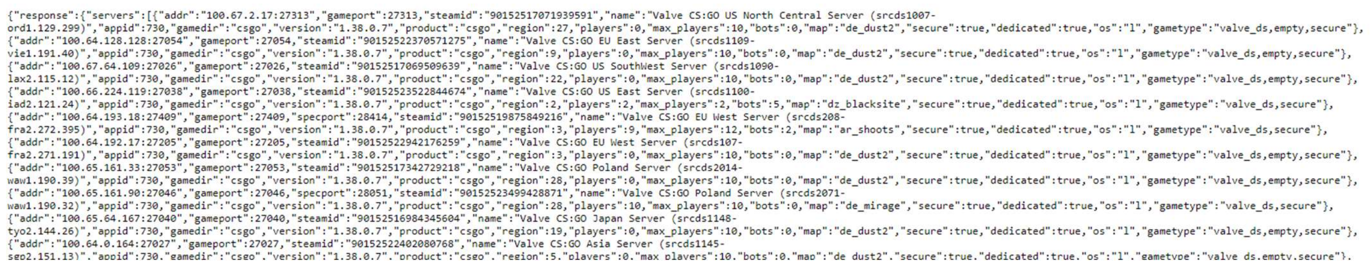
Type	Hexadecimal code	Meaning
Message Type	0x31	Query request for all servers
Region Code	0xFF	Return a list of servers from all regions
IP PORT	0x30 2E 30 2E 30 2E 30 3a 30 00	Translates to IP Port: 0.0.0.0:48 Establish a socket with the master server for response and to establish future connections. Like DNS, must start with 0.0.0.0 as your IP address.
Filter	0x00	No filter restrictions

The client will establish a socket connection and send a UDP packet to Steam's master servers. The packet shown above would be the default query packet used to get a list of all available servers from the master server[6]. Like other UDP packets sent through a socket connection, the packet sent and converted between hexadecimal and int32. The master server will process the packet and extract the byte array to read the request message[7]. The master server will reply with its own message in the form of a UDP Reply packet. Originally, this response was much simpler in that it simply responds with a UDP header of FF FF FF FF 66 0A with the rest of the message being a series of IP Addresses and their respective ports being used. However, the protocol was changed into an undocumented version[8].

As of December 2016, Steam has migrated the Master Server Query Protocol and integrated it into their Steam client. This means that the Steam client must send its request directly to the Steam platform to receive a list of server IPs. This new process is not documented, and the UDP headers used weren't found. As a result, any attempts to detect the

[6] https://gaming.stackexchange.com/questions/298348/the-server-list-from-source-engine-games-only-show-very-few-servers
[7] https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol/Parsing_packet
[8] https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol#Reply_format

new version of the Master Server Protocol have been unsuccessful. In addition, any attempts to use the legacy protocol even through Steam's Python API[9] have proven to be unsuccessful because all master servers have been updated to reject all UDP queries from clients[10], and all legacy games have been updated to use the new Master Server Query format[11]. This was part of Valve's on-going updates to prevent malicious users from abusing Steam's services to initiate DDoS reflection attacks and to prevent packet spoofing[12]. Currently, this protocol can be initiated only through the Steam client, Steam's publisher directory, or through the web client which requires an API key. This made prevented me from recording and documenting the current version and the legacy version of the Master Server Query through Wireshark.



```
{"response":{"servers":[{"addr":"100.67.2.17:27313","gameport":27313,"steamid":"90152517071939591","name":"Valve CS:GO US North Central Server (srcds1007-
ord1.129.299)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":27,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.64.128.128:27054","gameport":27054,"steamid":"90152522370571275","name":"Valve CS:GO EU East Server (srcds1109-
vie1.191.40)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":9,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.67.64.109:27026","gameport":27026,"steamid":"90152517069509639","name":"Valve CS:GO US SouthWest Server (srcds1090-
lax2.115.12)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":22,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.66.224.119:27038","gameport":27038,"steamid":"90152523522844674","name":"Valve CS:GO US East Server (srcds1100-
iad2.121.24)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":2,"players":2,"max_players":2,"bots":5,"map":"dz_blacksite","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,secure"},
{"addr":"100.64.193.18:27409","gameport":27409,"specport":28414,"steamid":"90152519875849216","name":"Valve CS:GO EU West Server (srcds208-
fra2.272.395)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":3,"players":9,"max_players":12,"bots":2,"map":"ar_shoots","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,secure"},
{"addr":"100.64.192.17:27205","gameport":27205,"steamid":"90152522942176259","name":"Valve CS:GO EU West Server (srcds107-
fra2.271.191)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":3,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.65.161.33:27053","gameport":27053,"steamid":"90152517342729218","name":"Valve CS:GO Poland Server (srcds2014-
waw1.190.39)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":28,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.65.161.90:27046","gameport":27046,"specport":28051,"steamid":"90152523499428871","name":"Valve CS:GO Poland Server (srcds2071-
waw1.190.32)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":28,"players":10,"max_players":10,"bots":0,"map":"de_mirage","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,secure"},
{"addr":"100.65.64.167:27040","gameport":27040,"steamid":"90152516984345604","name":"Valve CS:GO Japan Server (srcds1148-
tyo2.144.26)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":19,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
{"addr":"100.64.0.164:27027","gameport":27027,"steamid":"90152522402080768","name":"Valve CS:GO Asia Server (srcds1145-
sgp2.151.13)","appid":730,"gamedir":"csgo","version":"1.38.0.7","product":"csgo","region":5,"players":0,"max_players":10,"bots":0,"map":"de_dust2","secure":true,"dedicated":true,"os":"l","gametype":"valve_ds,empty,secure"},
```

This is the Master Server Query response output received when sending a request via Web API:
https://api.steampowered.com/IGameServersService/GetServerList/v1/?access_token=XXX

Regardless of the method used, the Master Server Query Protocol will provide the client with a list of server IP Addresses, the port used based on any filters for a specific game. However, this may not be enough information to assist a user in choosing a server[13]. The client may want to request more information such as latency, game type, custom server settings, and

[9] https://github.com/ValvePython/steam/blob/7ddb7f3eb9d951692a1a01fcdc73638a3034fbc0/steam/game_servers.py
[10] https://www.mail-archive.com/hlds_linux@list.valvesoftware.com/msg67998.html
[11] https://www.mail-archive.com/hlds_apps@list.valvesoftware.com/msg00978.html
[12] https://steamcommunity.com/discussions/forum/14/2989789048633291344/
[13] https://steam.readthedocs.io/en/latest/api/steam.game_servers.html

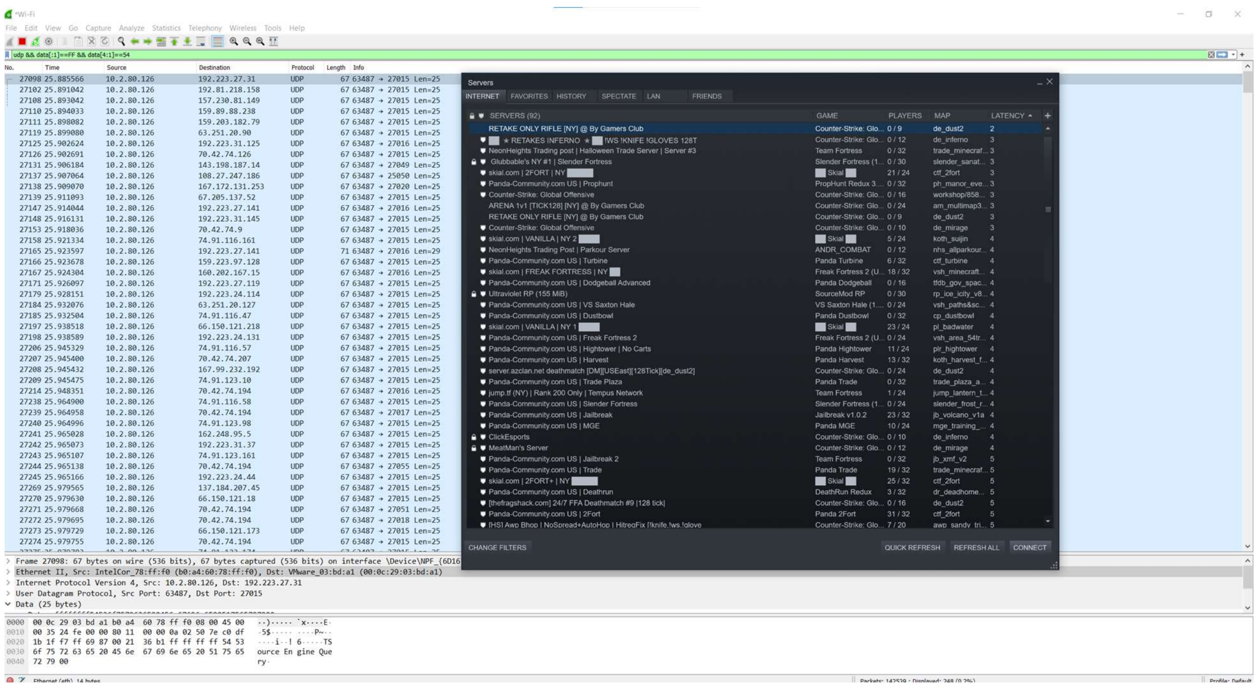
what other players might be on the server. Since our client now has a list of IP Addresses, the client can simply ask each servers directly for more information, leading to an additional set of server query protocols.

Getting more details about every server

In order to acquire further details regarding a server, the client will forge what the UDP header labels as the Source Engine Query packet. While this packet is labelled as such, the protocol uses 3-5 different UDP packets, each with a different header used to request a specific set of information from the server. Out of the 5 query packets, 2 have been documented as obsolete and been deprecated from all Source Engine games.

- **A2S_INFO**
    - A UDP packet sent to request basic server information and directed at a specific server of an IP Address.
    - This packet is sent to every server found when a game client requests a list of all available servers.
    - Requested details include player count, server name, game in progress, and the player count of that server.
    - Starts with FF FF FF FF 54 53 to mark itself as an A2S_INFO Request packet
    - Includes a 4-byte challenge number to prevent packet spoofing and reflection attacks[14].

[14] https://steamcommunity.com/discussions/forum/14/2989789048633291344/



```
udp && data[:1]==FF && data[4:1]==54

No.   Time      Source        Destination      Protocol  Length Info
27098 25.885566 10.2.80.126   192.223.27.31    UDP       67 63487 → 27015 Len=25
27102 25.891042 10.2.80.126   192.81.218.158   UDP       67 63487 → 27015 Len=25
27108 25.893042 10.2.80.126   157.230.81.149   UDP       67 63487 → 27015 Len=25
27110 25.894033 10.2.80.126   159.89.88.238    UDP       67 63487 → 27015 Len=25
27111 25.898082 10.2.80.126   159.203.182.79   UDP       67 63487 → 27015 Len=25
27119 25.899080 10.2.80.126   63.251.20.90     UDP       67 63487 → 27015 Len=25
27125 25.902624 10.2.80.126   192.223.31.125   UDP       67 63487 → 27016 Len=25
27126 25.902691 10.2.80.126   70.42.74.126     UDP       67 63487 → 27015 Len=25
27131 25.906184 10.2.80.126   143.198.187.14   UDP       67 63487 → 27049 Len=25
27137 25.907064 10.2.80.126   108.27.247.186   UDP       67 63487 → 25050 Len=25
27138 25.909070 10.2.80.126   167.172.131.253  UDP       67 63487 → 27020 Len=25
27139 25.911093 10.2.80.126   67.205.137.52    UDP       67 63487 → 27015 Len=25
27147 25.914044 10.2.80.126   192.223.27.141   UDP       67 63487 → 27016 Len=25
27148 25.916131 10.2.80.126   192.223.31.145   UDP       67 63487 → 27015 Len=25
27153 25.918036 10.2.80.126   70.42.74.9       UDP       67 63487 → 27015 Len=25
27158 25.921334 10.2.80.126   74.91.116.161    UDP       67 63487 → 27015 Len=25
27165 25.923597 10.2.80.126   192.223.27.141   UDP       71 63487 → 27016 Len=29
27166 25.923678 10.2.80.126   159.223.97.128   UDP       67 63487 → 27015 Len=25
27167 25.924304 10.2.80.126   160.202.167.15   UDP       67 63487 → 27016 Len=25
27171 25.926097 10.2.80.126   192.223.27.119   UDP       67 63487 → 27015 Len=25
27179 25.928151 10.2.80.126   192.223.24.114   UDP       67 63487 → 27015 Len=25
27184 25.932076 10.2.80.126   63.251.20.127    UDP       67 63487 → 27015 Len=25
27185 25.932504 10.2.80.126   74.91.116.47     UDP       67 63487 → 27015 Len=25
27197 25.938518 10.2.80.126   66.150.121.218   UDP       67 63487 → 27015 Len=25
27198 25.938589 10.2.80.126   192.223.24.131   UDP       67 63487 → 27015 Len=25
27206 25.945329 10.2.80.126   74.91.116.57     UDP       67 63487 → 27015 Len=25
27207 25.945400 10.2.80.126   70.42.74.207     UDP       67 63487 → 27015 Len=25
27208 25.945432 10.2.80.126   167.99.232.192   UDP       67 63487 → 27015 Len=25
27209 25.945475 10.2.80.126   74.91.123.10     UDP       67 63487 → 27015 Len=25
27214 25.948351 10.2.80.126   70.42.74.194     UDP       67 63487 → 27016 Len=25
27238 25.964900 10.2.80.126   74.91.116.58     UDP       67 63487 → 27015 Len=25
27239 25.964958 10.2.80.126   70.42.74.194     UDP       67 63487 → 27017 Len=25
27240 25.964996 10.2.80.126   74.91.123.98     UDP       67 63487 → 27015 Len=25
27241 25.965028 10.2.80.126   162.248.95.5     UDP       67 63487 → 27015 Len=25
27242 25.965073 10.2.80.126   192.223.31.37    UDP       67 63487 → 27015 Len=25
27243 25.965107 10.2.80.126   74.91.123.161    UDP       67 63487 → 27015 Len=25
27244 25.965138 10.2.80.126   70.42.74.194     UDP       67 63487 → 27055 Len=25
27245 25.965166 10.2.80.126   192.223.24.44    UDP       67 63487 → 27015 Len=25
27269 25.979565 10.2.80.126   137.184.207.45   UDP       67 63487 → 27015 Len=25
27270 25.979630 10.2.80.126   66.150.121.18    UDP       67 63487 → 27015 Len=25
27271 25.979668 10.2.80.126   70.42.74.194     UDP       67 63487 → 27051 Len=25
27272 25.979695 10.2.80.126   70.42.74.194     UDP       67 63487 → 27018 Len=25
27273 25.979729 10.2.80.126   66.150.121.173   UDP       67 63487 → 27015 Len=25
27274 25.979755 10.2.80.126   70.42.74.194     UDP       67 63487 → 27015 Len=25

> Frame 27098: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface \Device\NPF_{6D16
> Ethernet II, Src: IntelCor_78:ff:f0 (b0:a4:60:78:ff:f0), Dst: VMware_03:bd:a1 (00:0c:29:03:bd:a1)
> Internet Protocol Version 4, Src: 10.2.80.126, Dst: 192.223.27.31
> User Datagram Protocol, Src Port: 63487, Dst Port: 27015
v Data (25 bytes)

0000  00 0c 29 03 bd a1 b0 a4  60 78 ff f0 08 00 45 00   ··)·····`x····E·
0010  00 35 24 fe 00 00 80 11  00 00 0a 02 50 7e c0 df   ·5$·········P~··
0020  1b 1f f7 ff 69 87 00 21  36 b1 ff ff ff ff 54 53   ····i··!6·····TS
0030  6f 75 72 63 65 20 45 6e  67 69 6e 65 20 51 75 65   ource Engine Que
0040  72 79 00                                           ry·
```

Servers

SERVERS (92)	GAME	PLAYERS	MAP	LATENCY
RETAKE ONLY RIFLE [NY] @ By Gamers Club	Counter-Strike: Glo...	0 / 9	de_dust2	2
★ RETAKES INFERNO ★ !WS !KNIFE !GLOVES 128T	Counter-Strike: Glo...	0 / 12	de_inferno	3
NeonHeights Trading post | Halloween Trade Server | Server #3	Team Fortress	0 / 32	trade_minecraf...	3
Glubbable's NY #1 | Slender Fortress	Slender Fortress (1...	0 / 30	slender_sanat...	3
skial.com | 2FORT | NY	Skial	21 / 24	ctf_2fort	3
Panda-Community.com US | Prophunt	PropHunt Redux 3....	0 / 32	ph_manor_eve...	3
Counter-Strike: Global Offensive	Counter-Strike: Glo...	0 / 16	workshop/858...	3
ARENA 1v1 [TICK128] [NY] @ By Gamers Club	Counter-Strike: Glo...	0 / 24	am_multimap3...	3
RETAKE ONLY RIFLE [NY] @ By Gamers Club	Counter-Strike: Glo...	0 / 9	de_dust2	3
Counter-Strike: Global Offensive	Counter-Strike: Glo...	0 / 10	de_mirage	3
skial.com | VANILLA | NY 2	Skial	5 / 24	koth_suijin	4
NeonHeights Trading Post | Parkour Server	ANDR_COMBAT	0 / 12	nhs_allparkour...	4
Panda-Community.com US | Turbine	Panda Turbine	6 / 32	ctf_turbine	4
skial.com | FREAK FORTRESS | NY	Freak Fortress 2 (U...	18 / 32	vsh_minecraft...	4
Panda-Community.com US | Dodgeball Advanced	Panda Dodgeball	0 / 16	tfdb_gov_spac...	4
Ultraviolet RP (155 MiB)	SourceMod RP	0 / 30	rp_ice_icity_v8...	4
Panda-Community.com US | VS Saxton Hale	VS Saxton Hale (1....	0 / 24	vsh_paths&sc...	4
Panda-Community.com US | Dustbowl	Panda Dustbowl	0 / 32	cp_dustbowl	4
skial.com | VANILLA | NY 1	Skial	23 / 24	pl_badwater	4
Panda-Community.com US | Freak Fortress 2	Freak Fortress 2 (U...	0 / 24	vsh_area_54tr...	4
Panda-Community.com US | Hightower | No Carts	Panda Hightower	11 / 24	plr_hightower	4
Panda-Community.com US | Harvest	Panda Harvest	13 / 32	koth_harvest_f...	4
server.azclan.net deathmatch [DM][USEast][128Tick][de_dust2]	Counter-Strike: Glo...	0 / 24	de_dust2	4
Panda-Community.com US | Trade Plaza	Panda Trade	0 / 32	trade_plaza_a...	4
jump.tf (NY) | Rank 200 Only | Tempus Network	Team Fortress	1 / 24	jump_lantern_t...	4
Panda-Community.com US | Slender Fortress	Slender Fortress (1...	0 / 24	slender_frost_r...	4
Panda-Community.com US | Jailbreak	Jailbreak v1.0.2	23 / 32	jb_volcano_v1a	4
Panda-Community.com US | MGE	Panda MGE	10 / 24	mge_training_...	4
ClickEsports	Counter-Strike: Glo...	0 / 10	de_inferno	4
MeatMan's Server	Counter-Strike: Glo...	0 / 12	de_mirage	4
Panda-Community.com US | Jailbreak 2	Team Fortress	0 / 32	jb_xmf_v2	5
Panda-Community.com US | Trade	Panda Trade	19 / 32	trade_minecraf...	5
skial.com | 2FORT+ | NY	Skial	25 / 32	ctf_2fort	5
Panda-Community.com US | Deathrun	DeathRun Redux	3 / 32	dr_deadhome...	5
[thefragshack.com] 24/7 FFA Deathmatch #9 |128 tick|	Counter-Strike: Glo...	0 / 16	de_dust2	5
Panda-Community.com US | 2Fort	Panda 2Fort	31 / 32	ctf_2fort	5
[HS] Awp Bhop | NoSpread+AutoHop | HitregFix [!knife,!ws,!glove	Counter-Strike: Glo...	7 / 20	awp_sandy_tri...	5

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
INTERNET FAVORITES HISTORY SPECTATE LAN FRIENDS
CHANGE FILTERS
QUICK REFRESH
REFRESH ALL
CONNECT

Whenever a Source Engine game or the Steam client has to display a list of server, it will send an A2S_INFO Request packet to every available server. The image shows a Wireshark capture shortly after requesting an unfiltered list of all available servers.

- **A2S_Player**
  - Sends a request to the server to obtain information about the players currently in that server's game session
  - This packet is sent whenever the game client requests more information regarding a specific server.
  - Starts with 0xFF FF FF FF 55 and includes the string "Source Engine Query\0" as part of its header.
- **A2S_Rules (Couldn't be found, possibly integrated into another protocol)**
  - Sends a request to the server about any special settings that a server might have[15]
  - These settings would include descriptions such as if the server is officially hosted by Valve Software Co., is the server using a custom game mode, is the server marked as being under Valve's VAC Anti-Cheat software, and any other special settings that could have been implemented by the server owner
  - Starts with 0x56 in its header
- **A2A_PING (Obsolete, deprecated)**
  - Merely a ping request used to check how much latency is between the user and the server host. This information is acquired when sending an A2S_INFO request or during the Master Server Query protocol

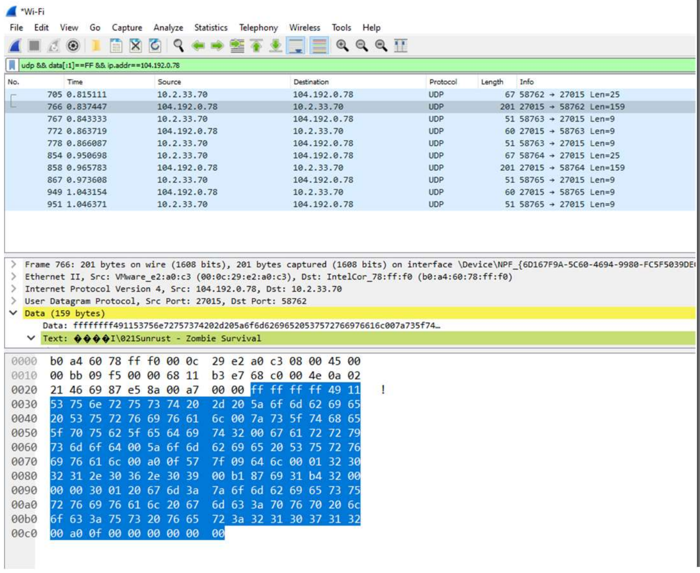
[15] https://developer.valvesoftware.com/wiki/ConVar

    - When trying to test for A2A_PING messages on Wireshark, only A2S_INFO packets were sent. This means that the ping function has been integrated with A2S_INFO.
    - Declared obsolete by Valve employees as of December 27, 2012, and is no longer used.
- **A2S_SERVERQUERY_GETCHALLENGE (Obsolete for most games)**
    - While we're using UDP to send and receive request and response packets, these messages are tailored towards specific clients and each user needs to have their conversation labelled to identify the client machine.
    - Whenever a user first connects to a server, the user would have to request a challenge number from the server[16].
    - That challenge number is how the user identifies himself when sending requests to the server[17].
    - This UDP packet is how the client would request a challenge number when speaking to a specific server to identify itself.
    - This format is no longer used for most games. Instead, clients will send an A2S_PLAYER or an A2S_RULES packet with an empty challenge number.

The server will receive the request and send what is labelled as the Source Multi-packet Response Format. Because we're sending it to one client and this response packet is also written in UDP which can be delayed or sent out of order, the packet header must be modified such that it can provide details and information regarding the server's connection to that specific client. This means that the UDP header must include an ID number assigned to that specific user and label to identify the packet order. This ID number is labelled as the Challenge Number. The server will randomly assign a Challenge Number to the client to identify that particular user's conversation[18]. This challenge number is attached at the end of all server reply packets which are sent based on the type of request packet received.

[16] https://store.steampowered.com/oldnews/422?l=english
[17] https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol#Challenge_response
[18] https://steamcommunity.com/discussions/forum/14/2989789048633291344/

- If the request packet was an A2S_INFO packet, it returns an A2S_INFO Response packet providing a list of server details and settings established by the server.

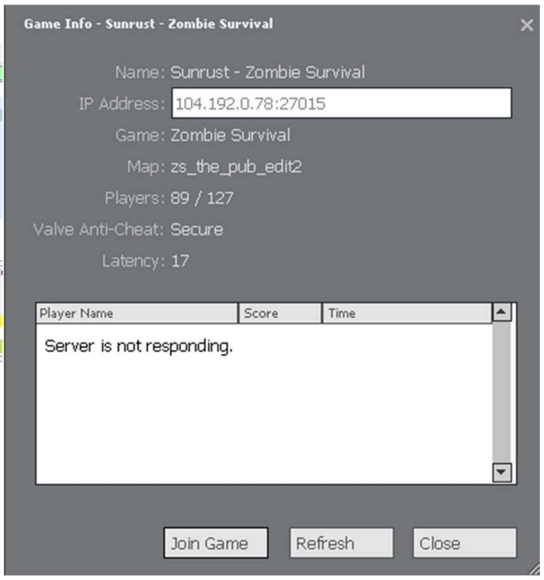

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
udp && data[:1]==FF && ip.addr==104.192.0.78

No.  Time      Source        Destination   Protocol Length Info
 705 0.815111  10.2.33.70    104.192.0.78  UDP       67 58762 → 27015 Len=25
 766 0.837447  104.192.0.78  10.2.33.70    UDP      201 27015 → 58762 Len=159
 767 0.843333  10.2.33.70    104.192.0.78  UDP       51 58763 → 27015 Len=9
 772 0.863719  104.192.0.78  10.2.33.70    UDP       60 27015 → 58763 Len=9
 778 0.866087  10.2.33.70    104.192.0.78  UDP       51 58763 → 27015 Len=9
 854 0.950698  10.2.33.70    104.192.0.78  UDP       67 58764 → 27015 Len=25
 858 0.965783  104.192.0.78  10.2.33.70    UDP      201 27015 → 58764 Len=159
 867 0.973608  10.2.33.70    104.192.0.78  UDP       51 58765 → 27015 Len=9
 949 1.043154  104.192.0.78  10.2.33.70    UDP       60 27015 → 58765 Len=9
 951 1.046371  10.2.33.70    104.192.0.78  UDP       51 58765 → 27015 Len=9

> Frame 766: 201 bytes on wire (1608 bits), 201 bytes captured (1608 bits) on interface \Device\NPF_{6D167F9A-5C60-4694-9980-FC5F5039DE
> Ethernet II, Src: VMware_e2:a0:c3 (00:0c:29:e2:a0:c3), Dst: IntelCor_78:ff:f0 (b0:a4:60:78:ff:f0)
> Internet Protocol Version 4, Src: 104.192.0.78, Dst: 10.2.33.70
> User Datagram Protocol, Src Port: 27015, Dst Port: 58762
v Data (159 bytes)
    Data: ffffffff491153756e72757374202d205a6f6d626965205375727669766616c007a735f74…
  v Text: ����I\021Sunrust - Zombie Survival

0000  b0 a4 60 78 ff f0 00 0c  29 e2 a0 c3 08 00 45 00
0010  00 bb 09 f5 00 00 68 11  b3 e7 68 c0 00 4e 0a 02
0020  21 46 69 87 e5 8a 00 a7  00 00 ff ff ff ff 49 11    !
0030  53 75 6e 72 75 73 74 20  2d 20 5a 6f 6d 62 69 65
0040  20 53 75 72 76 69 76 61  6c 00 7a 73 5f 74 68 65
0050  5f 70 75 62 5f 65 64 69  74 32 00 67 61 72 72 79
0060  73 6d 6f 64 00 5a 6f 6d  62 69 65 20 53 75 72 76
0070  69 76 61 6c 00 a0 0f 57  7f 09 64 6c 00 01 32 30
0080  32 31 2e 30 36 2e 30 39  00 b1 87 69 31 b4 32 00
0090  00 00 30 01 20 67 6d 3a  7a 6f 6d 62 69 65 73 75
00a0  72 76 69 76 61 6c 20 67  6d 63 3a 70 76 70 20 6c
00b0  6f 63 3a 75 73 20 76 65  72 3a 32 31 30 37 31 32
00c0  00 a0 0f 00 00 00 00 00  00
```



Name: Sunrust - Zombie Survival
IP Address: 104.192.0.78:27015
Game: Zombie Survival
Map: zs_the_pub_edit2
Players: 89 / 127
Valve Anti-Cheat: Secure
Latency: 17

Player Name	Score	Time
Server is not responding.		

Label for each part of the packet:

Hex Code	Meaning
0xff ff ff ff 49	49 labels this as a response packet
0x11	0001 0001 This is a public server, and it has VAC anti-cheat enabled
0x 53 75 6e 72 75 73 74 20 2d 20 5a 6f 6d 62 69 65 20 53 75 72 76 69 76 61 6c 00	Server Name: Sunrust – Zombie Survival
0x7a 73 5f 74 68 65 5f 70 75 62 5f 65 64 69 74 32 00	Map Name: zs_the_pub_edit2
0x67 61 72 72 79 73 6d 6f 64 00	Game Name: garrysmod
0x 5a 6f 6d 62 69 65 20 53 75 72 76 69 76 61 6c 00	Game Mode: Zombie Survival (This is what the server browser displays when looking at the server)
0x67 6d 3a 7a 6f 6d 62 69 65 73 75 72 76 69 76 61 6c 20	Server Tag gamemode is set to zombiesurvival (This is how the browser labels and sees the server's gamemode)
0x76	This server supports a maximum of 118 players (+ 9 bots)
0x65	There are 101 human players present in the game
0x80 00	This server supports up to a maximum of 128 players
0x 01 32 30 32 31 2e 30 36 2e 30 39 00	Date the server was created: 06/09/2021
0x 20 67 6d 63 3a 70 76 70	This game type is PVP
0x20 76 65 72 3a 32 31 30 37 31 32 00	Version Number: 210712

0x6c 6f 63 3a 75 73 20 76 65 72 3a 32 31 30 37 31 32 00	This server is located in the U.S.
0x00 00 00 00	End of packet

The packet response recorded during this game is different from the default format found on Valve's wiki page. The creator of this game appears to have changed the network protocol to better suit his game. Compared to the page describing the server response format, the following types were either not found or taken out in this version of the Source Engine:

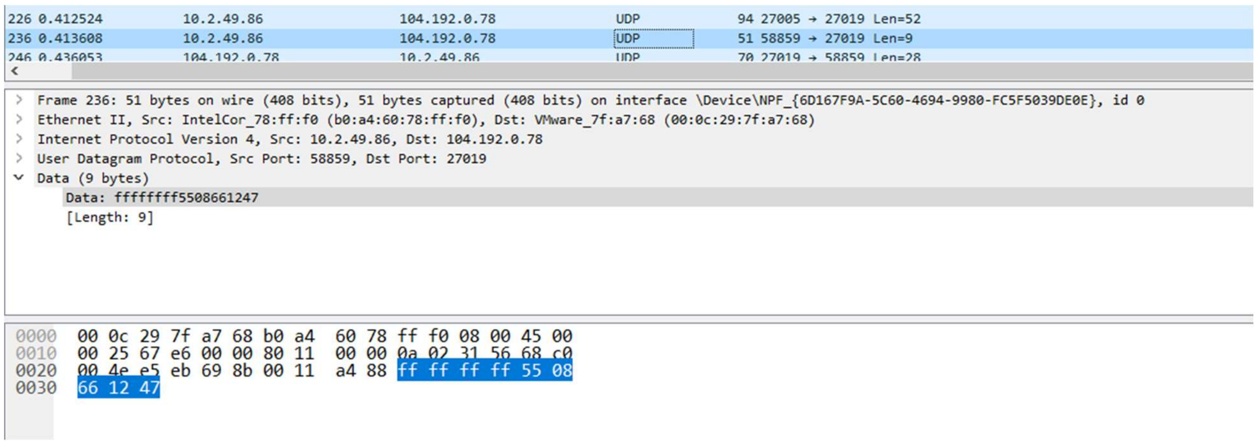
- ID: Gives the Steam App ID used to identify the type of game
  - Possibly removed because the game name is already included as part of the header. The ID was probably excess information.
- Folder – Identifies the folder where the game is located.
  - Information that isn't used or necessary when learning about the list of servers to join.
- Server Type – Identifies if the server is a dedicated server or a peer-to-peer server
  - Server browser has no setting or makes any distinction between connecting to a dedicated server or peer-to-peer server. They're all treated the same.
- Environment – Identifies the operating system this server is using.
  - Unnecessary information to learn about when connecting to a server.

It is not mentioned why these parts of the header were removed, but it could be that the headers mentioned above were considered unnecessary and were removed in later versions of the Source Engine or from this specific game.

A2S_Player Request packet:

When the game client presents a list of servers, it automatically sends a set of A2S_INFO queries to every available server to get the basic server information. A player might want to request more information for a specific server which might pertain to what settings a server has and what players are participating in the session. If a user wants to obtain more information regarding the players, the game client will send another UDP packet requesting the server to provide more regarding the other players. To observe this exchange, I joined an empty server and used Wireshark to record myself requesting the player count of the server I was in.

A2S_Player Request Packet:

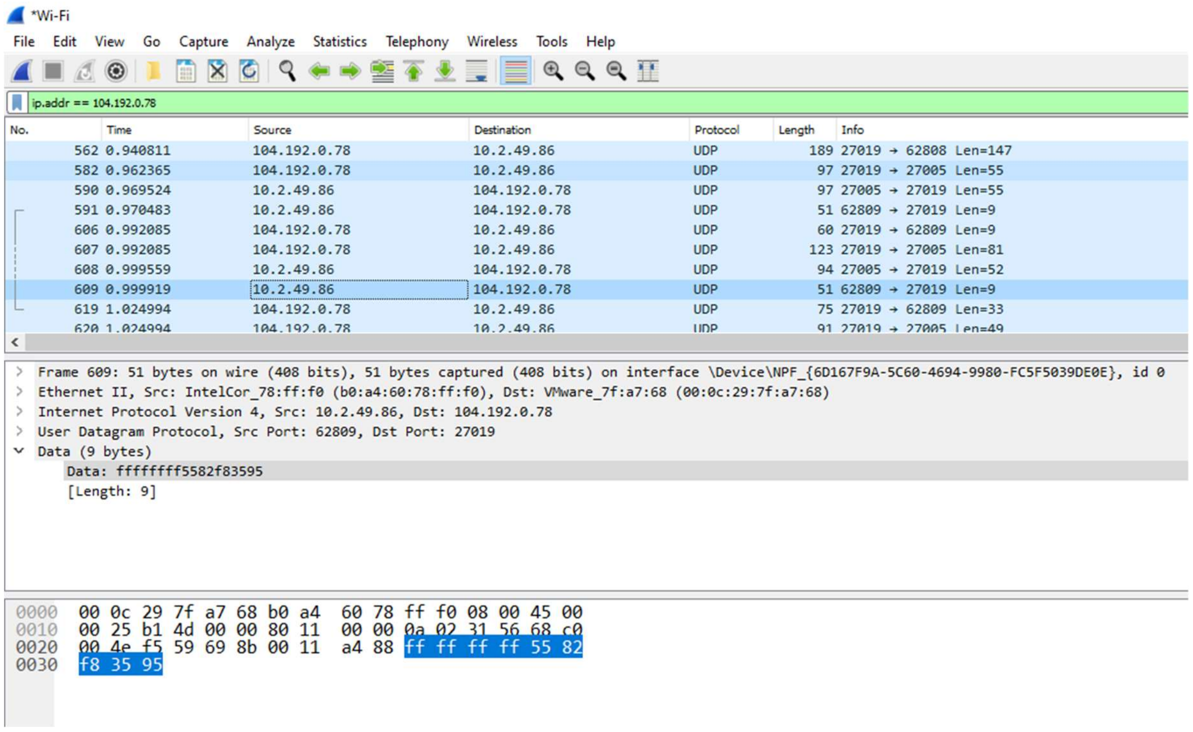


```
226 0.412524     10.2.49.86      104.192.0.78     UDP     94 27005 → 27019 Len=52
236 0.413608     10.2.49.86      104.192.0.78     UDP     51 58859 → 27019 Len=9
246 0.436053     104.192.0.78    10.2.49.86       UDP     70 27019 → 58859 Len=28

> Frame 236: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \Device\NPF_{6D167F9A-5C60-4694-9980-FC5F5039DE0E}, id 0
> Ethernet II, Src: IntelCor_78:ff:f0 (b0:a4:60:78:ff:f0), Dst: VMware_7f:a7:68 (00:0c:29:7f:a7:68)
> Internet Protocol Version 4, Src: 10.2.49.86, Dst: 104.192.0.78
> User Datagram Protocol, Src Port: 58859, Dst Port: 27019
v Data (9 bytes)
    Data: ffffffff5508661247
    [Length: 9]

0000  00 0c 29 7f a7 68 b0 a4  60 78 ff f0 08 00 45 00
0010  00 25 67 e6 00 00 80 11  00 00 0a 02 31 56 68 c0
0020  00 4e e5 eb 69 8b 00 11  a4 88 ff ff ff ff 55 08
0030  66 12 47
```

A2S_Player Response Packet:



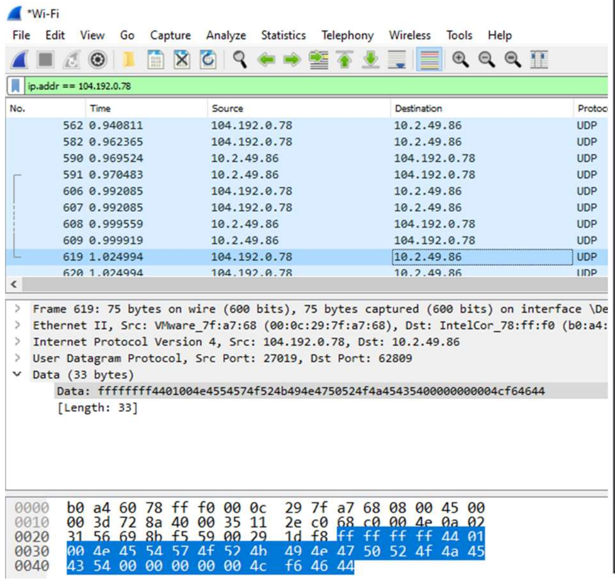
*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 104.192.0.78

```
No.    Time        Source          Destination     Protocol  Length  Info
   562 0.940811    104.192.0.78    10.2.49.86      UDP          189  27019 → 62808 Len=147
   582 0.962365    104.192.0.78    10.2.49.86      UDP           97  27019 → 27005 Len=55
   590 0.969524    10.2.49.86      104.192.0.78    UDP           97  27005 → 27019 Len=55
   591 0.970483    10.2.49.86      104.192.0.78    UDP           51  62809 → 27019 Len=9
   606 0.992085    104.192.0.78    10.2.49.86      UDP           60  27019 → 62809 Len=9
   607 0.992085    104.192.0.78    10.2.49.86      UDP          123  27019 → 27005 Len=81
   608 0.999559    10.2.49.86      104.192.0.78    UDP           94  27005 → 27019 Len=52
   609 0.999919    10.2.49.86      104.192.0.78    UDP           51  62809 → 27019 Len=9
   619 1.024994    104.192.0.78    10.2.49.86      UDP           75  27019 → 62809 Len=33
   620 1.024994    104.192.0.78    10.2.49.86      UDP           91  27019 → 27005 Len=49

> Frame 609: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \Device\NPF_{6D167F9A-5C60-4694-9980-FC5F5039DE0E}, id 0
> Ethernet II, Src: IntelCor_78:ff:f0 (b0:a4:60:78:ff:f0), Dst: VMware_7f:a7:68 (00:0c:29:7f:a7:68)
> Internet Protocol Version 4, Src: 10.2.49.86, Dst: 104.192.0.78
> User Datagram Protocol, Src Port: 62809, Dst Port: 27019
v Data (9 bytes)
    Data: ffffffff5582f83595
    [Length: 9]

0000  00 0c 29 7f a7 68 b0 a4  60 78 ff f0 08 00 45 00
0010  00 25 b1 4d 00 00 80 11  00 00 0a 02 31 56 68 c0
0020  00 4e f5 59 69 8b 00 11  a4 88 ff ff ff ff 55 82
0030  f8 35 95
```

The Source A2S_PLAYER request packet sent when requesting more server information. Note that the data after 0x55 represents the Challenge Number.

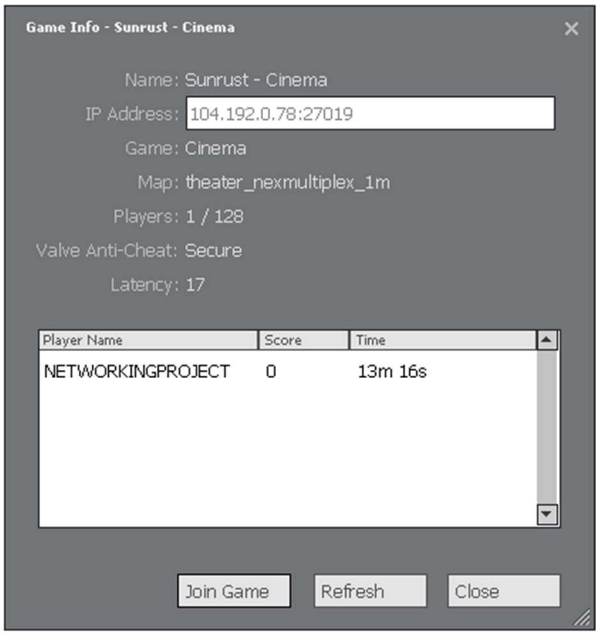

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 104.192.0.78

No.   Time      Source         Destination    Protoc
  562 0.940811  104.192.0.78   10.2.49.86     UDP
  582 0.962365  104.192.0.78   10.2.49.86     UDP
  590 0.969524  10.2.49.86     104.192.0.78   UDP
  591 0.970483  10.2.49.86     104.192.0.78   UDP
  606 0.992085  104.192.0.78   10.2.49.86     UDP
  607 0.992085  104.192.0.78   10.2.49.86     UDP
  608 0.999559  10.2.49.86     104.192.0.78   UDP
  609 0.999919  10.2.49.86     104.192.0.78   UDP
  619 1.024994  104.192.0.78   10.2.49.86     UDP
  620 1.024994  104.192.0.78   10.2.49.86     UDP

> Frame 619: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \De
> Ethernet II, Src: VMware_7f:a7:68 (00:0c:29:7f:a7:68), Dst: IntelCor_78:ff:f0 (b0:a4:
> Internet Protocol Version 4, Src: 104.192.0.78, Dst: 10.2.49.86
> User Datagram Protocol, Src Port: 27019, Dst Port: 62809
v Data (33 bytes)
    Data: ffffffff4401004e4554574f524b494e4750524f4a45435400000000004cf64644
    [Length: 33]

0000  b0 a4 60 78 ff f0 00 0c  29 7f a7 68 08 00 45 00
0010  00 3d 72 8a 40 00 35 11  2e c0 68 c0 00 4e 0a 02
0020  31 56 69 8b f5 59 00 29  1d f8 ff ff ff ff 44 01
0030  00 4e 45 54 57 4f 52 4b  49 4e 47 50 52 4f 4a 45
0040  43 54 00 00 00 00 00 4c  f6 46 44
```



Name: Sunrust - Cinema
IP Address: 104.192.0.78:27019
Game: Cinema
Map: theater_nexmultiplex_1m
Players: 1 / 128
Valve Anti-Cheat: Secure
Latency: 17

Player Name	Score	Time
NETWORKINGPROJECT	0	13m 16s

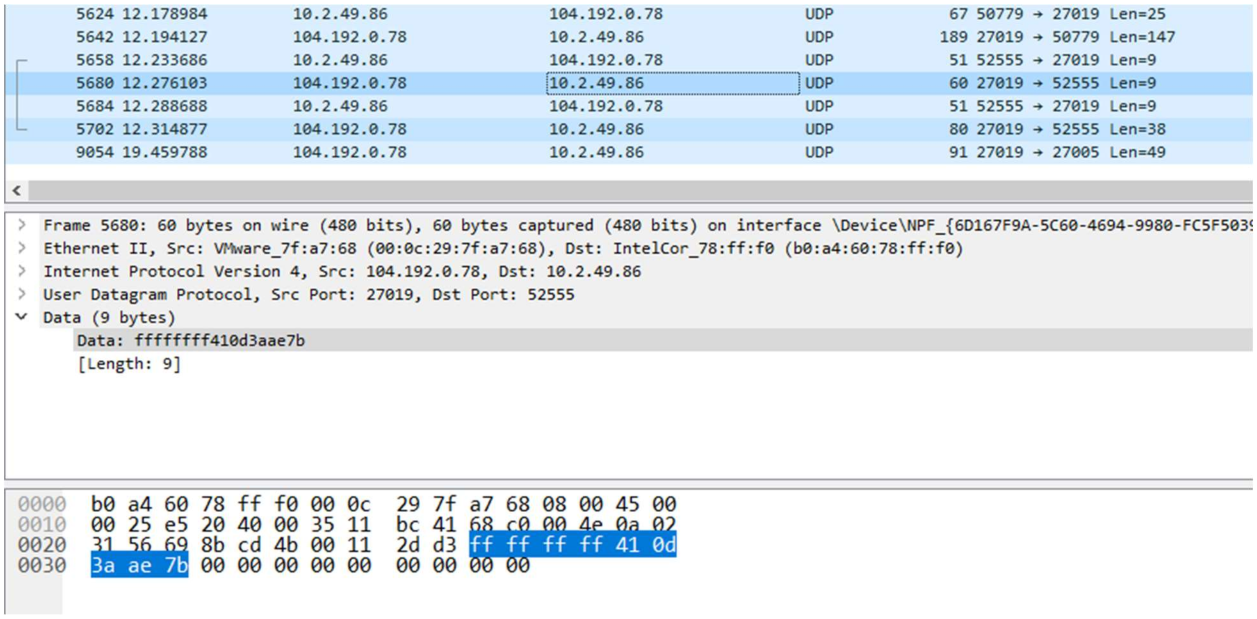
Hexadecimal Code	Translated Meaning
0x44	A2S_Player Reponse packet
0x 01 00	There is 1 player on the server (Only me)
0x 4e 45 54 57 4f 52 4b 49 4e 47 50 52 4f 4a 45 43 54 00	User on the server: NETWORKINGPROJECT (My in-game name)
0x00 00	This player is at index 0
0x00 00	This player has 0 points
0x4c f6 46 44	Unknown: Presumably tied to the duration the player has spent on the server

- As shown in the two images above,
    - The response packet I received provided my client with details regarding my name, my score, and possibly my current time spent recording on the server
    - Asking for server info and refreshing server info would have my client send an A2S_PLAYER Request packet and an A2S_INFO packet for server information and player information
    - Both packets were sent separately and received their own respective response packet.

A2S_PING & A2S_RULES:

With A2S_PING being obsolete, no UDP packets containing the matching header were found despite using ping commands in-game and sending ping requests directly to a server.

However, while tracking packets, messages containing the A2S_SERVERQUERY_GETCHALLENGE header were found. This was in spite that no request packets were sent out of or found on Wireshark. This is because the A2S_INFO and A2S_RULES query may also initiate a challenge request number if needed. However, this is unusual as this feature was supposed to be deprecated, and the game I was testing my Wireshark capture on was not on the list of games that still supported this feature[19].



```
5624 12.178984   10.2.49.86     104.192.0.78   UDP    67 50779 → 27019 Len=25
5642 12.194127   104.192.0.78   10.2.49.86     UDP   189 27019 → 50779 Len=147
5658 12.233686   10.2.49.86     104.192.0.78   UDP    51 52555 → 27019 Len=9
5680 12.276103   104.192.0.78   10.2.49.86     UDP    60 27019 → 52555 Len=9
5684 12.288688   10.2.49.86     104.192.0.78   UDP    51 52555 → 27019 Len=9
5702 12.314877   104.192.0.78   10.2.49.86     UDP    80 27019 → 52555 Len=38
9054 19.459788   104.192.0.78   10.2.49.86     UDP    91 27019 → 27005 Len=49

> Frame 5680: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{6D167F9A-5C60-4694-9980-FC5F503
> Ethernet II, Src: VMware_7f:a7:68 (00:0c:29:7f:a7:68), Dst: IntelCor_78:ff:f0 (b0:a4:60:78:ff:f0)
> Internet Protocol Version 4, Src: 104.192.0.78, Dst: 10.2.49.86
> User Datagram Protocol, Src Port: 27019, Dst Port: 52555
v Data (9 bytes)
     Data: ffffffff410d3aae7b
     [Length: 9]

0000  b0 a4 60 78 ff f0 00 0c  29 7f a7 68 08 00 45 00
0010  00 25 e5 20 40 00 35 11  bc 41 68 c0 00 4e 0a 02
0020  31 56 69 8b cd 4b 00 11  2d d3 ff ff ff ff 41 0d
0030  3a ae 7b 00 00 00 00 00  00 00 00 00
```

Data captured	Translated Meaning
0x41	This is an A2S_SERVERQUERY_GETCHALLENGE packet
0x 0d 3a ae 8b 00 00 00 00 00 00 00 00 00	This represents the challenge number used to identify the specific conversation that my game client has with the server

While the Source Challenge Response packet was still obtainable, A2S_RULES was an unknown packet that couldn't be found despite any attempts to send a query a server. Using Steam's Python API to directly send an A2S_RULES request yields no response, this protocol

[19] https://developer.valvesoftware.com/wiki/Server_queries#A2S_SERVERQUERY_GETCHALLENGE

wasn't found in any Source Engine games being tested, and there is no publicly documented method of acquiring an A2S_RULES packet, so it is uncertain if the protocol has been retired or has a more obscure method needed to request for server rules.

However, in the figure showcasing the master server query protocol via web API, every server displayed a set of server configurations such as game, game mode, security settings, and map name. These additional settings fit the description of what A2S_RULES was supposed to provide as the Master Server Query protocol itself was initially designed to provide only the list of IP Addresses and port numbers used by every server. It is likely that A2S_RULES was incorporated with the Master Server Query protocol. Regardless, A2S_INFO and A2S_PLAYER are the two main packets sent from a client to a server, and these two packets are most often used to acquire the necessary info displayed the game. When presented with this information, the user will have the necessary information to decide which server to join in, fulfilling the goals of Steam's Source Engine Networking protocols.

Brandon Vo

References and discrepancies

https://steamcommunity.com/discussions/forum/14/2989789048633291344/

https://steamcommunity.com/discussions/forum/14/2974028351344359625/

- These were community posts released by one of Valve's Software Developers, Fletcher Dunn who is responsible for managing Steam's networking software.
- This post describes changes made to the A2S_INFO protocol to improve its anti-spoofing resistance.
- These posts describe the changes made to the packet header which are used today along with a description for the purpose for some of the packet changes and explained some of the packet's anti-spoofing and DDoSing countermeasures.

https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol

https://developer.valvesoftware.com/wiki/Server_queries

- This is the main source of information for this research paper as it has the most information regarding the
- These wiki sites were created and curated by Valve Software
- However, the site is managed and edited by community members which may lead to certain inaccuracies and discretion regarding reliability of the information provided.
  - This is further exacerbated by the issue that not all articles were up to date with posts and discussions taking place over 5 years ago.
- Regardless, the site is referenced multiple times by Valve's employees on their GitHub site for reference, and the employees also manage the site and answer any discrepancies on the discussion page as well which can verify the site's credibility.

https://steamapi.xpaw.me/#IGameServersService/GetServerList

- Used to initiate a Master Server Query protocol via web API.
- A token was acquired from Steam's web developer request site which was used to send the API request.
- However, any attempts to trace the requests packets on the web yielded no results because there was not enough information regarding what packet to be looking for.
- This site only provides a list of what each web command does and what it provides, but there is no documentation regarding how it works so this site wasn't very useful for information gathering.

https://github.com/ValveSoftware/GameNetworkingSockets

- Because Valve has their engine open to the public, they also made their networking software open-sourced on Github.
- The Github page was used mainly to read the additional notes left behind by the developers explaining what some of their protocols do.

- However, any attempts to use their protocol manually send request packets on Python were unsuccessful because Valve's servers were updated to no longer accept packets that didn't originate from the Steam client.
    - This is possibly related to their anti-spoofing and reflection prevention measures.