

Timeout Analysis, Troubleshooting and Notification in Real Time Bidding Advertising System with Implementation

Jayant Kumar

Director of Platform Integration and Architecture (Senior Software Architect), Bidtellect Inc., Delray Beach, USA

Abstract Programmatic Advertising ecosystem needs to have system which can show Ad to an online User within few milliseconds like 200 to 300 ms. An active user on internet don't stay on the website or App forever. So, any missed opportunity to show Ad is missed revenue. Timeout is a major concern for all programmatic AdTech players in Digital Advertising ecosystem. This paper talks about a researched approach to identify Timeout, analyze, and troubleshoot the timeout issue and notify the involved partners with all information as quickly as possible.

Keywords RTB, Programmatic Advertising, Timeout Notification, Ad Server, Programmatic Exchange

1. Introduction

Whenever user goes online either on a website or App, Ad Server or SSP gets all the available information about user and website/App, send the same to all Demand Partner Server.

Most of publishers work with multiple Supply partners and have restriction of getting an Ad within 100 ms to 300 ms. SSP performs some optimization and send the information to all connected Demand partner server. Demand partner server need to respond within allowed period which can be 100ms to 200ms.

In the entire work flow if any involved entity Demand partner or SSP couldn't respond within allowed time lose the opportunity of showing Ad to the user. This situation is known as Timeout.

Hence, Timeout is a lost revenue opportunity. It's very important to regularly keep check on Timeout, analyze the same and notify involved parties with troubleshooting information.

Challenge is that every SSP or AdServer have multiple Data center and Timeout can be a problem on a specific Data center with a specific Demand Partner. So, we need a method to analyze Timeout at every Data center and troubleshoot the same with Demand partner having timeout issue.

It's very often said in programmatic world that Lost impression is a lost revenue. Throughout the Article, we would be using few keywords which are as below.

SSP – A supply-side platform is a piece of software used to sell advertising in an automated fashion. SSPs are most often used by online publishers to help them sell display, video, and mobile ads. [1]

DSP – A system that allows buyers of digital advertising inventory to manage multiple ad exchange and data exchange accounts through one interface., Real-time bidding for displaying online advertising takes place within the ad exchanges, and by utilizing a DSP, marketers can manage their bids for the banners and the pricing for the data that they are layering on to target their audiences. Much like Paid Search, using DSPs allows users to optimize based on set Key Performance Indicators such as effective Cost per Click (eCPC), and effective Cost per Action (eCPA). [2]

OpenRTB Specification - The mission of the OpenRTB project is to spur growth in Real-Time Bidding (RTB) marketplaces by providing open industry standards for communication between buyers of advertising and sellers of publisher inventory. There are several aspects to these standards including but not limited to the actual real-time bidding protocol, information taxonomies, offline configuration synchronization, and many more. This document specifies a standard for the Real-Time Bidding Interface.

OpenRTB has multiple version of real time bidding specifications like Version 2.0, 2.1, 2.2, 2.3, 2.5. [8]

Impression - An impression (in the context of online advertising) is when an ad is fetched from its source, and is countable. Whether the ad is clicked is not considered. Each time an ad is fetched, it is counted as one impression. [3]

* Corresponding author:

jayant42@gmail.com (Jayant Kumar)

Published online at <http://journal.sapub.org/computer>

Copyright © 2017 Scientific & Academic Publishing. All Rights Reserved

2. Research Method

If we look at the Real-time bidding ecosystem, Timeout can happen because of two Reason.

1. Due to Latency in Network
2. Due to Latency in DSP System/Server

If we analyze the Network Time between SSP Server and DSP Server, we can find the Network travel/Connection time. If that's high, it means there is problem with Network.

In the case Network latency is high, we can analyze the time between each gateway by looking at the path that was used to communicate from SSP Server to DSP Server, to find the route/gateway which is exactly causing the Network latency. As it could be due to issue in ISP provider of SSP or ISP provider of DSP.

We can use **tracert** [6] or **tracert** [7] like tools which Internet Control Message Protocol (ICMP) Echo Request messages to the destination with incrementally increasing Time to Live (TTL) field values. The path displayed is the list of near-side router interfaces of the routers in the path between a source host and a destination.

If there is No Latency in Network, Then We need to analyze the full time taken by DSP to respond.

As it might be possible that, DSP system is taking a lot of time to process an Ad Request.

DSP generally respond with either an empty response or a valid Ad response. Empty response means that DSP is not able to show Ad for the requested Ad request.

Look at the time taken by DSP for both kind of responses, we can find patterns as below.

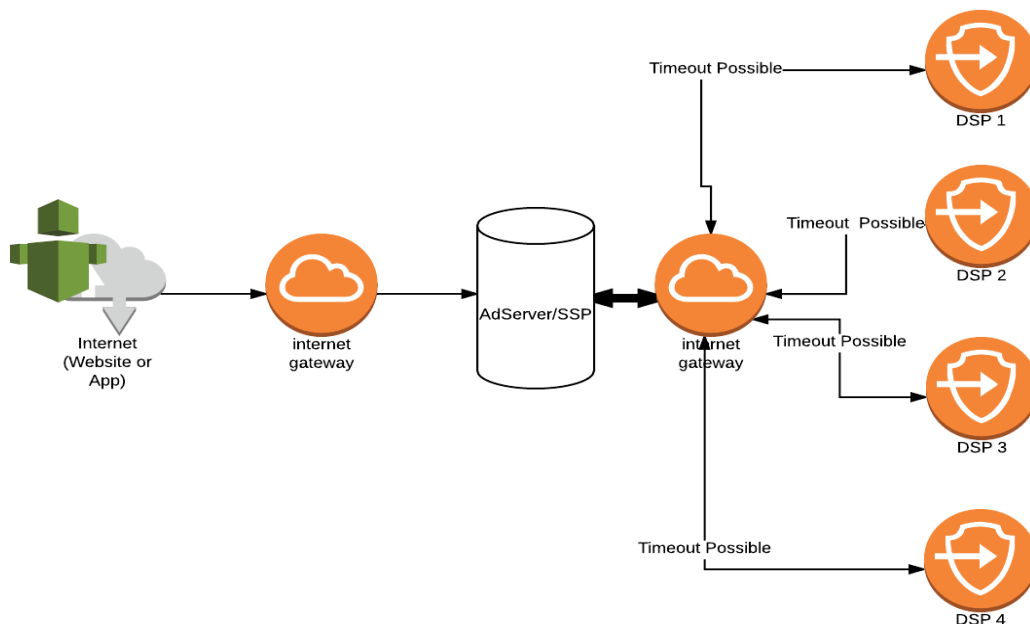
- Time taken by DSP for both empty and valid response takes more time than allowed.
- Time taken by DSP for empty response is low but Its very high for responding with a valid Ad response

In below Diagram, the flow of data from the time User get online till the Ad is served to the user is demonstrated.

Below is the sequence of events happen to show an Ad on a digital platform.

1. User goes online (Either on a website or App or smart device)
2. Ad Server gets all the information from the device and send the same to SSP
3. SSP send Ad requests to all connected DSPs
4. DSP process the request and return with an empty or valid Ad response
5. SSP only entertain Ad response from DSPs who can bid within the allowed time frame otherwise they are considered timed out.
6. SSP perform internal auction, if applicable
7. SSP send Ad response back to Ad Server
8. Ad Server also has time restriction and entertain response from a SSP who can bid within the time constrain
9. Ad server renders the Ad on the Digital content (website or App)

Flow Diagram of Ad request:



3. Analysis and Implementation

Every SSP or AdServer can have multiple Data Center. Every data centers can have different timeout restriction. Every DSP also have respective Data Centers to avoid Latency.

We can apply the research method to analyze timeout root cause from a specific data center of SSP to respective DSP endpoint or Data Center.

We can create a service to look at timeout stats periodically and analyze timeout using above method for the Data Center which has high latency.

Let's define the timeout for ever Data center and use traceroute to find if Network has an issue. We should look at every gateway and highlight that path which is causing high latency.

Send an email with the Timeout stats with traceroute [6] samples from SSP server to DSP endpoints highlighting the gateway which is causing high latency.

Traceroute Example:

Below highlighted gateways have high latency. This information can be used to notify the ISP provider to fix the network routing.

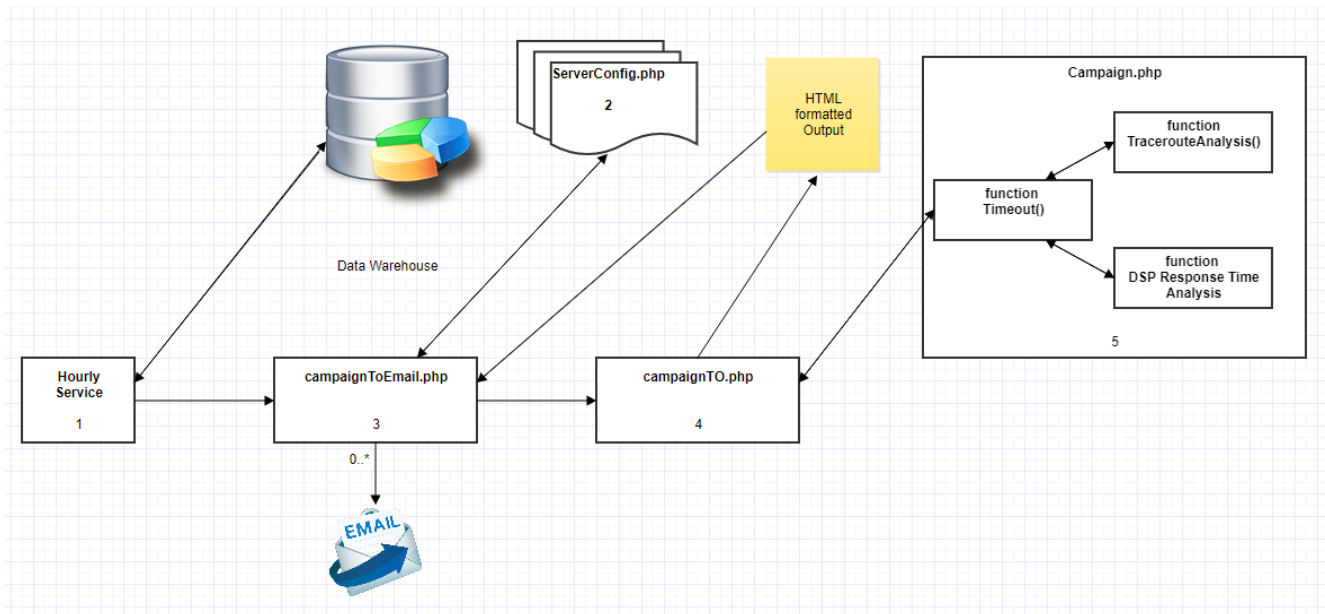
```
traceroute to 207.198.110.38 (207.198.110.38), 30 hops max, 60 byte packets
 1 72.37.167.2 (72.37.167.2) 0.250 ms 0.238 ms 0.211 ms
 2 162.248.17.17 (162.248.17.17) 0.328 ms 0.322 ms 0.296 ms
 3 xe-2-2-0.mpr4.sfo3.us.above.net (208.185.125.245) 0.248 ms 0.192 ms 0.162 ms
 4 ae11.cr2.sjc2.us.above.net (64.125.24.225) 28.819 ms 28.794 ms 28.740 ms
 5 ae10.mpr4.sjc7.us.above.net (64.125.31.74) 2.991 ms 5.271 ms 10.036 ms
 6 ae11.mpr3.sjc7.us.above.net (64.125.28.1) 25.481 ms 7.881 ms 45.392 ms
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

If there is no Network Latency, then we should look at response time of DSP as mentioned in Research method.

We can do the same by using curl command. We can use curl command to send Ad request to DSP having high timeout from the Server at Datacenter which has high timeout.

Analyze the various kind of responses received from DSP as mentioned in research method to find the pattern and highlight the ones which is causing high timeout.

Gather all those information for every DSP having high timeout and send the same in an email.

Implementation Flow Diagram:**Pseudo Code Implementation using php:****Assumption:**

We are using Linux environment. All the DSP information is stored in MySQL.

We already have keyless SSH access to servers which would be used to perform troubleshooting and Analysis.

1: Create a Service which runs on regular interval like hourly and find the DSP which are having high Timeout and execute **campaignToEmail.php** by passing respective DSP or campaign Id which has high timeout.

2: Create a config file to store IPs of servers for each data centre which is available to perform troubleshooting. Name the file as **ServerConfig.php**

Method Usage and Explanation:

Below method takes the Data Center Id as argument and return the list of servers for the specific Data center which can be used for troubleshooting.

Point to be Noted: We can have 100s or 1000s of server at a Data center, we should limit ourselves to specific set of servers to use for troubleshooting and analysis.

ServerConfig.php code snippet:

```

<?php
function GetServerList ($dc)
{
    $ServerList;
    switch ($dc)
    {
        case 1:
            $ServerList = array("Comma Separated list of Servers in a data center 1"); //data Center 1
            break;
        case 2:
            $ServerList = array("Comma Separated list of Servers in a data center 2"); //data Center 2
            break;
        case 3:
            $ServerList = array("Comma Separated list of Servers in a data center 3"); //data Center 3
    }
}

```

```

        break;
    case 4:
        $ServerList = array("Comma Separated list of Servers in a data center 4"); //data Center 4
        break;
    case 5:
        $ServerList = array("Comma Separated list of Servers in a data center 5"); //data Center 5
        break;
    default:
        throw new Exception ( 'Unlisted DC ID requested');
        break;
    }
    return($ServerList);
}
?>

```

3: Create a file **CampaignToEmail.php** which will take DSP Id or name as input and analyse Timeout for the same at all configured data centre and store the information in a html file.

Function Explanation:

It takes 3 arguments.

\$Task – Task name, TOD is the task name used for Timeout Detail

\$campaignID – Id of DSP Campaign or Account

\$CampaignName – Name of DSP Campaign or Account

It checks if user has permission to run the task. It executes **CampaignTo.php** for the requested task (which would TOD) and store the output in a temporary file.

It uses another helper method **SendMail.php** to send an email the respective DSP partner with the content of temporary file.

Temporary file has all information about the Timeout Analysis.

Pseudo Code snippet:

```

<?php
date_default_timezone_set("America/Los_Angeles");
if ($_GET) {
    $Task = $_GET['argument1'];
    $campaignID = $_GET['argument2'];
    $CampaignName = $_GET['argument3'];
} else {
    $Task = $argv[1];
    $campaignID = $argv[2];
    $CampaignName = (empty($argv[3]) ? '' : (strtoupper($Task) == "TRI" ? $argv[3] : '.'.$argv[3]));
}
$send = 1;
$out = 0;
$filename = $Task.mt_rand()."_".date("Ymd",strtotime( "-$1 days" ))."_".$campaignID.".html";
//echo $CampaignName;
if((strtoupper($Task) != "TOD" && (strtoupper($Task) == "HELP" )) //To print information about using this script or command
{
    //echo "Invalid Options\n";
    echo "Below are valid Options :\n";

    echo "TOD - Timeout , traceroute and all other additional info .\n Example : php CampaignTOEmail.php". ' "TOD" ' ' '
"5400" ' ' ' ' "x@mail.com" ' . "\n";
    exit;
}

```



```

$Task = (empty($argv[1]) ? " " : $argv[1]) ;
$campaignID = (empty($argv[2]) ? " " : $argv[2]) ;
}

function CampaignTOTask($Task,$campaignID,$html,$startdate,$enddate,$frontmsg)
{
    if((strtoupper($Task) != "TO" && strtoupper($Task) != "username" && strtoupper($Task) == "HELP" )
    {
        //echo "Invalid Options\n";
        echo "Below are valid Options :\n";
        echo "TO - Timeout of todays or yesterday whichever is available in system. \n Example : php
CampaignTO.php". " " "TO" " " "CampaignId" " " "0" " " \n";
        exit;
    }
    $user = exec('whoami');
    $campaign = new Campaign($campaignID); //Create Instance/Object of Campaign Class
    try
    {
        switch ($Task)
        {
            case "TO":
                $campaign->Timeout($html); //Call the method Timeout
                break;
            default:
                echo "\nInvalid Input";
                break;
        }
        echo "</br></br><p><font color=\".\".\"red\".\"\".\"><b>Note : This is a system generated email . Please contact
'x@mail.com' in case of any query or issue .</b></font></br><b> Data/Information mentioned above is not validated and thus should
be used just as an Information nothing else .<b> </p></br></br> <b>Thanks</br></b>";
    }
    Catch(Exception $e)
    {
        return(0);
    }
    return(1);
}

CampaignTOTask($Task,$campaignID,$html,$startdate,$enddate,$frontmsg);

?>

```

5.1: Traceroute Analysis

Below is the main method which can analyze **traceroute** [6] results executed from SSP server for the Demand Partner API URL.

Arguments to the Function:

\$url – Demand Partner API url,

\$dc- Data center Id

\$html=1 – 1 means we want to return html formatted output

\$all – 1 means that we want to show traceroute samples from all Servers which are configured for troubleshooting

```
function TracerouteAnalysis($url,$dc,$html=1,$all=0)
{
    $valid = 1; //Used to set if traceroute sample was valid
    $current; //To store string throughout the method
    $conntm; //To set connection timeout threshold for the specific Data Center
    $Zone;    //To Store Data Center Name
    $out; //To store string throughout the method
    try
    {
        $sList = GetServerList($dc); //Get list of Servers configured for the Data Center
    }
    Catch(Exception $e)
    {
        return;
    }
    //echo print_r($sList);
    if(count($sList) <= 0)
    {
        return;
    }
    switch($dc)           //Based on data Center , set the zone and connection timeout threshold
    {
        case 1:
            $Zone = 'Data Center 2';
            $conntmc = $this->DataCenter2;
            break;

        case 2:
            $Zone = 'DataCenter1';
            $conntmc = $this->DataCenter1;
            break;

        case 3:
            $Zone = 'DataCenter3 ';
            $conntmc = $this->DataCenter3;
            break;

        case 4:
            $Zone = 'DataCenter4';
            $conntmc = $this->DataCenter4;
            break;

        case 6:
            $Zone = 'DataCenter5';
            $conntmc = $this->DataCenter5;
            break;

        default:
            $Zone = $dc;
            $conntmc = 30;
    }
}
```



```

        break;
    }
    $p = strpos($url, 'http'); //parse DSP API as url
    $output = parse_url($url);
    if($p === false)
    {
        $path = $output['path']; //get the host of DSP API url
    }
    else
    {
        $path = $output['host']; // get the host of DSP API url
    }

    foreach($sList as $tserver)
    {
        $valid = 1;
        $cmdServer = 'ssh -q '.$tserver.' /sbin/ifconfig | grep 'eth1' -A 1 | grep 'inet addr' | awk -F: '{print $2}' | awk
'{print $1}';

        try
        {
            unset($outs);
            $serverIP = "";
            exec($cmdServer, $outs, $codes);
            if(sizeof($outs) == 1)
                $serverIP = $outs[0]; // get the IP of all servers which are configured for troubleshooting

        }
        Catch(Exception $e)
        {
            continue;
        }
        $cmd = 'ssh -q '.$tserver.' traceroute "'.$path.'"'; //run traceroute[6] using the IP of remort server for the host of
DSP API url

        try
        {
            exec($cmd, $out, $code);
        }
        Catch(Exception $e)
        {
            $valid = 0 ;
            continue;
        }
        if ($code)
        {
            continue;
        }

        $current = $out; //Current traceroute response
        $size = sizeof($out);
        $ct = $this->ConnectionTime($out); //Analyse the Traceroute response to find the connection time
        $conntmc = $ct; //set connection time which will be compared with threshold timeout to find if its
exceeding time limit

        unset($out); //Empty the out variable
        if($ct >= $conntmc && $all != 1)
        {
            break;

```

```

    }
    if($all == 1)          //If we need to display all Samples irrespective if the connection time is normal or high
    {
        if($html == 0) //print the Traceroute with final time
        {
            echo "\n$Zone:\n";
            echo "Server IP : $serverIP\n$ConnectionTime:$sconntm\nBelow is the Sample:\n";
            foreach($current as $t)
            {
                echo "$t\n";
            }
        }
        else //print the Traceroute with final time in html format highlighting Connection time in Red colour if it
        exceeds the threshold time
        {
            if($valid == 1)
            {
                if($sconntm >= $sconntmc)
                {
                    $st = "<b><font color='\"red\"'> Connection takes more than threshold value of $sconntmc ms.</font></b> ";
                }
                else if($sconntm >= ($sconntmc - ($sconntmc * .70)) and $sconntm < $sconntmc)
                {
                    $st = " Connection Time looks OK ";
                }
                else
                {
                    $st = " Connection Time looks good.";
                }
                echo "<br><b>$Zone:</b></br>";
                echo "Server IP : $serverIP<br><b>ConnectionTime</b>:$sconntm. $st<br>Below is the Sample.</br>";
                foreach($current as $t)
                {
                    echo "$t<br>";
                }
            }
        }
    }
}

if(isset($sconntm) == false || $sconntm == 0)
    $valid = 0;
if($all != 1) //print the last Traceroute sample with final time
{
    if($html == 0)
    {
        echo "\n$Zone:\n";
        echo "Server IP : $serverIP\nConnectionTime:$sconntm\nBelow is the Sample:\n";
        foreach($current as $t)
        {
            echo "$t\n";
        }
    }
    else
    {
        if($valid == 1)
        {
            if($sconntm >= $sconntmc)
            {
                $st = "<b><font color='\"red\"'> Connection takes more than threshold value of $sconntmc ms.
</font></b>";
            }
            else if($sconntm >= ($sconntmc - ($sconntmc * .70)) and $sconntm < $sconntmc)
            {
                $st = "Connection Time looks OK ";
            }
            else
            {
                $st = "Connection Time looks good.";
            }
        }
    }
}

```

```

        $st = "Connection Time looks good .";
        echo "</br><b>$Zone:</b></br>";
        echo "Server IP : $serverIP</br><b>ConnectionTime</b>:$conntm. $st</br>Below is the Sample.</br>";

        foreach($current as $t)
        {
            echo "$t</br>";
        }
    }
    else
    {
        //print_r($code);
        if(isset($current) == true)
        {
            foreach($current as $t)
            {
                echo "$t</br>";
            }
        }
    }
}
}
}
}

```

5.2: DSP Response Time Analysis:

There is a possibility that there is no issue in Network. Traceroute analysis will show good results in that case.

So, now we should send sample request to DSP API url.

Please note that most of DSPs use optimization and don't bid on every ad request. We need to send bid requests to get two types of Responses.

1. DSP bided with no or empty response

This will tell us the time, it took for a DSP when they choose not to bid. We should try to get sample of multiple such responses. Most of the time DSP should be able to decide early not to bid and response time should be relatively low.

2. DSP bided with an Ad response

This will tell us the time, it took for a DSP to process the Ad request and apply all internal decision rules and return an eligible Ad response. If there is a latency in a DSP engine or bottle neck situation, time will be high.

In the case we see the time in both types of response is very high, then it direct DSP to check their internal setup or network.

Please refer Open RTB 2.5 [7.0] specification to get sample OpenRTB Ad Requests.

Below is few sample curl call to send Bid requests to a DSP API URL.

Curl Sample to send an Ad request to a DSP:

```

curl -w '\n %{http_code},%{time_total},%{time_namelookup},%{time_connect},%{time_starttransfer} \n' -H "Content-Type: application/json" -X POST -d
'{"id": "7264466533724059265", "imp": [{"id": "1881962729045211389", "native": {"request": {"native": {"ver": "1", "layout": "3", "adunit": "0", "plcmcnt": "1", "assets": [{"id": "1", "required": "1", "title": {"len": "100"}, {"id": "6", "required": "0", "img": {"type": "3"}, {"id": "7", "required": "0", "img": {"type": "1"}, {"id": "2", "required": "0", "data": {"type": "2", "len": "300"}, {"id": "3", "required": "0", "data": {"type": "12", "len": "20"}, {"id": "4", "required": "0", "data": {"type": "1"}, {"id": "5", "required": "0", "data": {"type": "3"}, {"plcmtype": "0"}}, "ver": "1", "instl": "0", "secure": "0", "ext": {"appnexus": {"estimated_clear_price": "1.000000", "member_ad_profile_id": "27079"}}, "site": {"id": "1958422", "domain": "adnxs.net", "page": "http://dspcreativetesting.adnxs.net/bidtellect_native_test.html?ast_debug=true&ast_dongle=QWERTY&ast_debug_member=958"}, "device": {"ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36", "geo": {"country": "USA"}, "dnt": "0", "ip": "10.1.13.154", "make": "Unknown", "model": "Unknown", "os": "10.11 El Capitan", "language": "en", "connectiontype": "0", "user": {"id": "681745333480637551", "ext": {"appnexus": {"userdata_json": {"}}}}, "test": "0", "at": "2", "tmax":

```

```
2000,"wseat":["3603"],"cur":["USD"],"bcat":["APN-74","IAB8-18","NEX8-101","IAB8-5","APN-158","APN-159"],"ext":{"appnexus":{"seller_member_id": 958,"spend_protection":false}}}' "DSP End Point"
```

Curl Sample to send an Ad request to a DSP through a Remote Server:

```
ssh -q ServerIP 'curl -s -w"%{http_code},%{time_total},%{time_namelookup},%{time_connect},%{time_starttransfer}\n"
"DSP_END_POINT"?requestId=C8CD44A2-D266-4EA7-8E78-D89AC41CCBDD&adWidth=300&adHeight=250&uid=FAE5E7C7-38D6-4C99-
A247-D5CDB6EB1F7A&ip=74.46.94.196&adTagType=JavaScript&bidCurrency=USD&siteurl=http%3A%2F%2Fweatherbug.com&pageurl
=www.WeatherBug.com&refurl=www.WeatherBug.com&adId=37697&pubId=33588&siteId=33608&inIframe=1&uFoldPos=1&sFoldPo
s=0&freq=0&timezone=-6&screenResolution=1024x768&language=en-US&browser=Mozilla%2F4.0%20%28compatible%3B%20MSIE%
207.0%3B%20Windows%20NT%206.1%3B%20Trident%2F7.0%3B%20SLCC2%3B%20.NET%20CLR%202.0.50727%3B%20.NET%20CLR%2
03.5.30729%3B%20.NET%20CLR%203.0.30729%3B%20Media%20Center%20PC%206.0%3B%20.NET4.0C%3B%20.NET4.0E%29&platfor
m=1"
```

4. Conclusions

By Analyzing Network Latency and Response Time of DSP Server, we can troubleshoot and find root cause of Timeout in a Real Time Bidding System. If there is an issue with Network involving SSP's ISP provided or same for the DSP's ISP provider, they can proactively fix the problem. SSP or DSP can also use this information to perform Network peering or updating their DNS lookup if needed.

We can also find out if issue is within the DSP System and same can be notified to DSPs regularly through Sample Ad response time. They can use that information to fix the problem in their system.

Regularly monitoring, analyzing Timeout issue in real time bidding system can avoid loss of impression and hence can avoid loss of revenue.

REFERENCES

- [1] <https://digiday.com/media/wtf-supply-side-platform/>
- [2] https://en.wikipedia.org/wiki/Demand-side_platform
- [3] [https://en.wikipedia.org/wiki/Impression_\(online_media\)](https://en.wikipedia.org/wiki/Impression_(online_media))
- [4] <https://en.wikipedia.org/wiki/CURL>
- [5] <http://php.net/manual/en/intro-what-is.php>
- [6] <https://en.wikipedia.org/wiki/Traceroute>
- [7] <https://technet.microsoft.com/en-us/library/bb491018.aspx>
- [8] <https://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf>