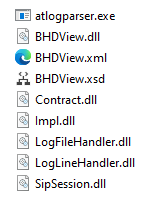
# ATLOGPARSER – a Parser for rvbeehd logs

Mike Luckham  
Jan 17, 2024

ATLOGPARSER is a command-line tool distributed in a dist folder or dist.7z



* atlogparser scans the application working directory for ‘view’ DLLs
* BHDView.dll processes BeeHD log files, it requires BHDView.xsd/.xml
* The other DLLs are loaded as well

Analyzing the output of the parser requires some technical knowledge and experience, but not as much as needed when looking at the raw log files.

The suggested approach is to analyze an rvbeehd.log file, determine the time range of interest, look at the Interpretation section of the SIP Session in the \_atparser.log file, and look at the charts on the corresponding Session\_ tab in the Excel workbook. For a more detailed dive, the Messages worksheets can be used to filter and examine the log lines in detail.

**Large Log Files and OutOfMemory exceptions** – very large log files (100,000 Kb or more) will cause an exception to be thrown and logged in the \_atparser.log file. No Excel file will be produced.

To eliminate this exception, divide the large log file into smaller ones less than 150,000 Kb – analyze the log files individually.

To decide where to split the file and avoid doing so in the middle of a call session, search for Disconnected->Terminated, then search for the next ResolverMgrRemoveResolver.

*CallLegChangeState - Call-leg xxxxxxx state changed: Disconnected->Terminated, (reason = Call Terminated)  
…  
ResolverMgrRemoveResolver - removing Resolver xxxxxx from list*

*<DIVIDE HERE>*

If you split in the wrong place, or if a log file naturally split in the middle of a call, the Session analysis of calls overlapping the point of split might not be complete and the Excel charts misleading. In \_atparser.log you see:  
  
 *\*\* SESSION \*\* WARNING Session does not start with INVITE*

*Call-id 74118918-4afee8a-8b0e-219d1678-243da8c0-13c4-65015.*

In such a case, where multiple rvbeehd\_hhmmss.log files are available, identify in which files the call starts and ends and move lines as needed so that one log file contains all the lines of call.

## Usage:

atlogparser.exe <.log|.txt|.zip|directory>

## Command-line Arguments:

* A directory name
  + The directory is recursively searched for files of pattern rvbeehd\*.log
* An archive name with extension .zip
  + The zip file is unzipped to a folder with the same name as the zip
  + The unzipped folder is then recursively searched, as for the directory name option
* A path whose filename ends with extension .log or .txt
  + The file is opened directly
  + Wildcards are not accepted

## Working Directory:

* Paths are relative to the current directory selected in the command window
* For convenience and to avoid long pathnames it’s recommended to set the current directory to where the AT log files to be analyzed are stored
* Output files (.log, .xml, .xsd, .xls) are stored in the working directory

## Outputs:

* An execution log file whose name is taken from the input log file:  
  + Example: Input (rvbeehd.log) output (rvbeehd\_atparser.log)
  + Example: Input (rvbeehd\_123.log) output (rvbeehd\_123\_atparser.log)
    - Contains all of the command-line output including execution information and SIP Session analyses
* Temporary XML and XSD files  
  + One pair BHDView.xml/xsd
    - Contains a record for each input log line from the input .log file
    - The XML is imported to Excel, then deleted
  + One or more pairs BHDView\_Session\_mmdd\_hhmm.xml/xsd
    - Each pair contains statistics for a single SIP Session
    - The XML pair is imported to Excel, then deleted
  + The temporary XML/XSD files are deleted after the Excel file is created
* Excel XLS
  + The name is taken from the input log file
    - Example: Input (rvbeehd.log) output (rvbeehd.xls)
    - Example: Input (rvbeehd\_123.log) output (rvbeehd\_123.xls)
  + Contains several worksheets:
    - Message, Message\_1, … each containing a maximum number of rows (100,000)
    - Session\_mmdd\_hhmm worksheets, one for each SIP session
    - TimeAdjustment … used to conveniently timeshift the log lines in the Message worksheets

## ATLOGPARSER Algorithm

* Zip file is unzipped
* Input directory is scanned to identify input log files
* Log files are read into memory line-by-line, and transcribed to XML form
* XML files are written (actually APPENDED if the XML file exists from a prior run)
* XML files are read and transcribed to Excel XLS worksheets
* XML files are deleted

## Large Input Logs and Out of Memory Exceptions

* Very large input logs (500,000+ lines) will result in out-of-memory exceptions, either while reading the input loglines and transcribing them to the XML files, or when loading the XML files into the Excel worksheets.
* Exceptions are recorded in the atlogparser.log
* When an exception occurs:
  + output data will be lost
  + most commonly the largest output file (BHDView.xml) is affected.
  + ATLOGPARSER may exit without completing the algorithm
  + This can leave XSD/XML files remaining in the working directory, and when the parser is run again it will READ THEM AND APPEND NEW DATA TO THEM
  + The correct procedure after an exception is to manually delete the XML/XSD files from the working directory

### Avoiding Out of Memory Exceptions

* Manually split the input BeeHD logs into smaller log files, and process them individually
* Take care not to split them in the middle of a SIP Session call

## Features of the ATPARSER.LOG

The areas of interest start with LOGSUMMARY.

*\*\* LOGSUMMARY \*\* Parser BHD, Filter rvbeehd\*.log, File , LineCount 418085, Version 4.7.24.8, Source WIndows, From 10/04/23 08:00:52.107, To 2023-01-01 00:00:00.000*

Here we have a SIP session call getting set up, and eventually ending with BYE. Notice the call took 5 seconds to be answered (the time between the initial INVITE and the confirmation by dsantoya with the OK message followed by ACK.

Note the call lasted 2 minutes 24 seconds – the time between the ACK and the BYE – and that BYE was sent indicating that this side initiated the call disconnection.

*\*\* SESSION \*\* Call-id 44e01e-5096f74f-4b17-235aa440-ef39a8c0-13c4-65015.*

*--sipmessages--*

*231004 08:21:35.061 00:00.000 INVITE 1 In <sip:192.168.57.239@192.168.57.239> 192.168.57.239 (1511 bytes)*

*231004 08:21:38.162 00:03.101 RINGING 1 Out <sip:10.255.254.111> (582 bytes)*

*231004 08:21:38.170 00:03.109 INVITE 1 In <sip:192.168.57.239@192.168.57.239> 192.168.57.239 (1511 bytes)*

*231004 08:21:38.170 00:03.109 INVITE 1 In <sip:192.168.57.239@192.168.57.239> 192.168.57.239 (1511 bytes)*

*231004 08:21:38.170 00:03.109 RINGING 1 Out <sip:10.255.254.111> (582 bytes)*

*231004 08:21:38.170 00:03.109 RINGING 1 Out <sip:10.255.254.111> (582 bytes)*

*231004 08:21:40.437 00:05.376 OK 1 Out <sip:10.255.254.111> dsantoyo (1498 bytes)*

*231004 08:21:40.516 00:05.455 OK 1 Out <sip:10.255.254.111> dsantoyo (1498 bytes)*

*231004 08:21:40.566 00:05.505 ACK 1 In <sip:192.168.57.239@192.168.57.239> (571 bytes)*

*231004 08:21:40.569 00:05.508 ACK 1 In <sip:192.168.57.239@192.168.57.239> (571 bytes)*

*231004 08:23:59.668 02:24.607 BYE 1 Out (559 bytes)*

*231004 08:23:59.754 02:24.693 BYE 1 Out (559 bytes)*

*231004 08:23:59.911 02:24.850 BYE 1 Out (559 bytes)*

*231004 08:24:00.223 02:25.162 BYE 1 Out (559 bytes)*

*231004 08:24:00.849 02:25.788 BYE 1 Out (559 bytes)*

*231004 08:24:01.080 02:26.019 OK 1 In (544 bytes)*

*231004 08:24:01.080 02:26.019 OK 1 In (544 bytes)*

*231004 08:24:01.080 02:26.019 OK 1 In (544 bytes)*

*231004 08:24:01.080 02:26.019 OK 1 In (544 bytes)*

*231004 08:24:01.081 02:26.020 OK 1 In (544 bytes)*

Following the SIP messages display is some interpretation of what happened.  
  
The largest message side (1511 bytes) is listed here – the network MTU must be at least that big, or network routers all along the network must support large-UDP-packet-reassembly otherwise large messages will be completely lost.  
  
If there were problems such as the call not being answered, the interpretation would state that.

*--interpretation--*

*Largest INVITE: 1511 bytes*

*Largest OK: 1511 bytes*

*231004 08:21:35.061 00:00.000 [TELLER] Incoming call request was received at machine.*

*INFO Call-id 44e01e-5096f74f-4b17-235aa440-ef39a8c0-13c4-65015*

*INFO The Inviter <sip:192.168.57.239@192.168.57.239> 192.168.57.239*

*INFO The Responder <sip:10.255.254.111> dsantoyo*

*INFO The Responder has a private IP address behind a NAT or VPN.*

*231004 08:21:38.162 00:03.101 Started RINGING and informed the remote machine.*

*INFO <sip:10.255.254.111> dsantoyo*

*231004 08:21:40.437 00:05.376 Accepted the call and informed the remote party to finalize establishing the connection.*

*231004 08:21:40.569 00:05.508 Success (Connected) - local and remote parties are connected, audio and video can flow.*

*231004 08:23:59.668 02:24.607 Call is ended, sent BYE to inform the remote party.*

*231004 08:24:01.080 02:26.019 Disconnected normally with BYE.*

*INFO Total time connected 02:20.511*

Following the interpretation are a list of events for each of the four media streams (audio in, audio out, video in, video out). The events will follow a general pattern where the microphone, speaker, camera are opened at the beginning of the call and closed at the end. The UDP network transmission rates are continually negotiated during a call by both ends of the BeeHD conversation – in an effort to maintain timely and quality audio-video – there are events listed each time such an adjustment is made.

*--statistics--*

*Audio-Incoming Transfer Rate statistics not available - enable MediaLogOptions|MediaSessionFilters = DEBUG*

*Good packets 14288, lost 185, average packet loss 1%, peak packet loss 11%*

*08:00:53.798 speaker-open speakers (2- logi usb headset)*

*08:00:56.361 audio-volume-normal 50% of max*

*speaker-on set (0- speakers (2- logi usb headset)) 100cb9b0*

*08:00:57.574 to 08:00:58.746 2 audio-volume-normal 50% of max*

*08:21:36.625 speaker-open speakers (2- logi usb headset)*

*08:21:37.458 speaker-on set (0- speakers (2- logi usb headset)) 15067680*

*08:21:37.460 speaker-open speakers (2- logi usb headset)*

*08:21:37.461 speaker-on set (0- speakers (2- logi usb headset)) 150b3258*

*08:21:37.477 audio-volume-normal 50% of max*

*08:21:37.516 speaker-open speakers (2- logi usb headset)*

*08:21:37.518 speaker-on set (0- speakers (2- logi usb headset)) 150b3258*

A call with poor network conditions will contain a lot of adjustments and errors in the input streams. A call with good network quality will not have very many of these, if any.

*08:23:17.053 audio-quality Audio QOS - lost 7 out of 660 (1.06%), LastSeq# 14734, Good 653, jitter 120 msec.*

*08:23:17.092 audio-receiver-ok*

*08:23:17.193 audio-receiver-quality Audio QOS - lost 1, LastSeq# 14741*

*08:23:26.532 to 08:23:26.652 3 audio-quality Audio QOS - lost 7 out of 479 (1.46%), LastSeq# 15208, Good 472, jitter 120 msec.*

*08:23:26.674 to 08:23:26.753 2 audio-receiver-quality Audio QOS - lost 1, LastSeq# 15215*

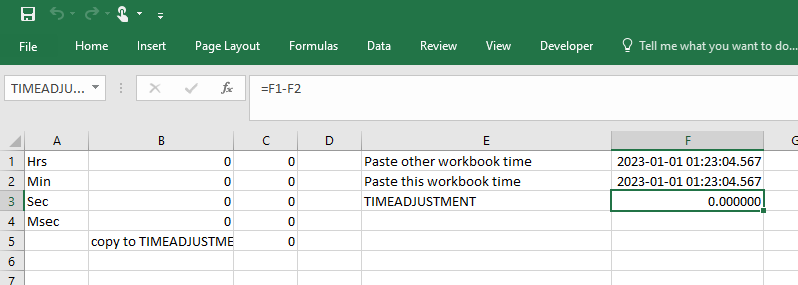
Following the statistics section there may be a CSV listing of the time buckets. These are the same as listed in the Excel spreadsheet Session\_ worksheets.

*\*\* CSV Time Buckets*

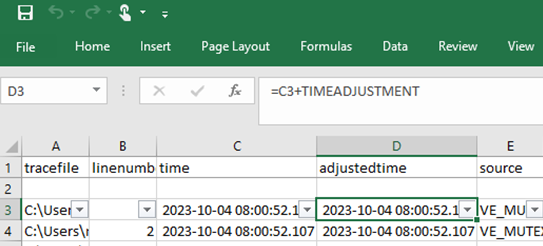
*239,,,,normal,,,,,,,,43000,43000,0,0,0,0,0,1145902,259397,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,56,65,0,0,0,0,0,0,0,0,0*

## Features of the Excel Spreadsheet

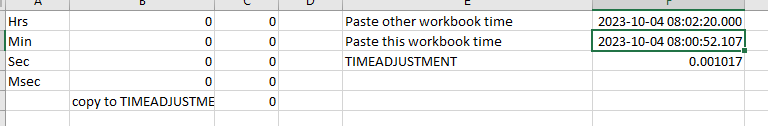
#### TimeAdjustment worksheet



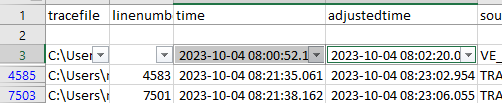
* Cell F3 is a named cell “TIMEADJUSTMENT”
* Each of the other worksheets has a time column and an adjustedtime column
* The cells of the adjustedtime column add the TIMEADJUSTMENT from F3 to the time value



* The intended use is when comparing Excel workbooks from two log files – one from the ATM terminal and one from the Teller.  
  + Identify a common action such as the INVITE being sent on one side and INVITE being received on the other side.
  + Choose one Excel workbook to adjust the time. Usually it will be the one with the least reliable clock synchronization (ie: the teller)
  + Copy the INVITE ‘time’ from this workbook to cell F2 (Paste this workbook time)
  + Copy the INVITE ‘time’ value from the other worksheet to cell F1 (Paste other workbook time)

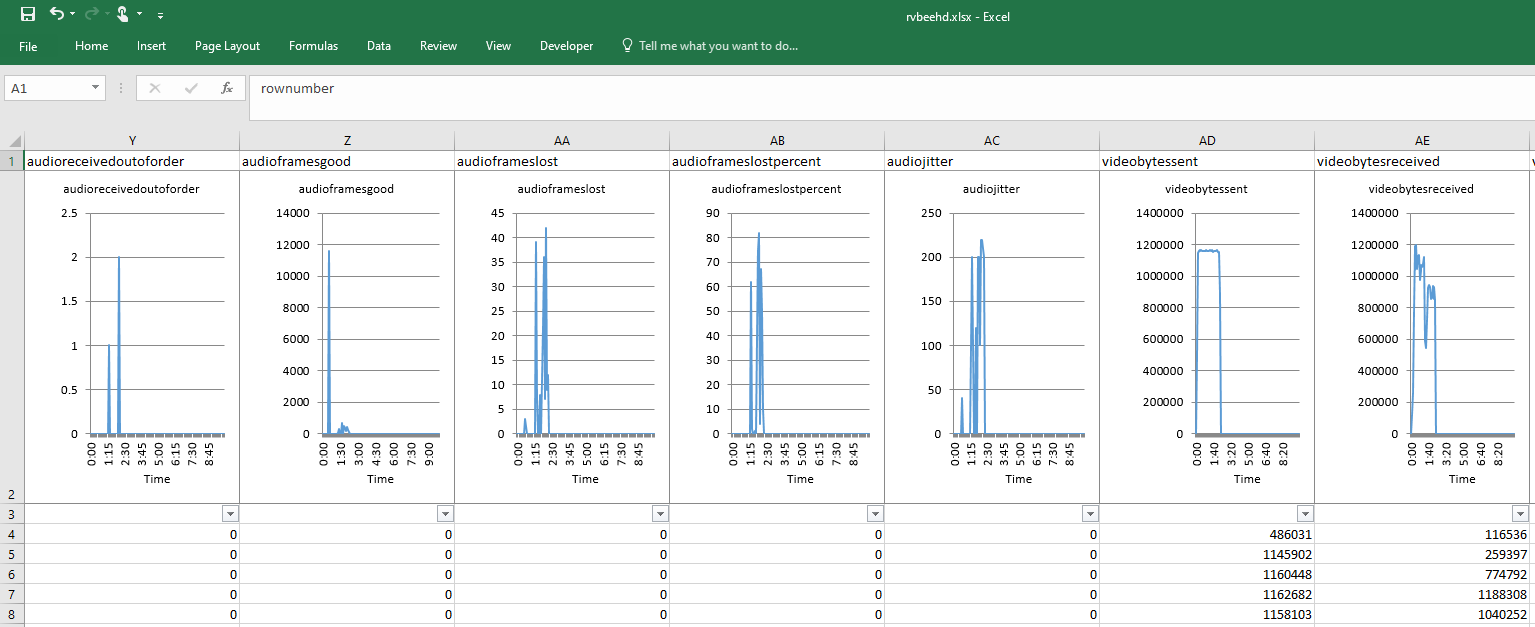


* + Referring now to the Messages or Sessions worksheets, you will find the adjustedtime values have changed, allowing an easier side-by-side comparison of ATM and teller logs

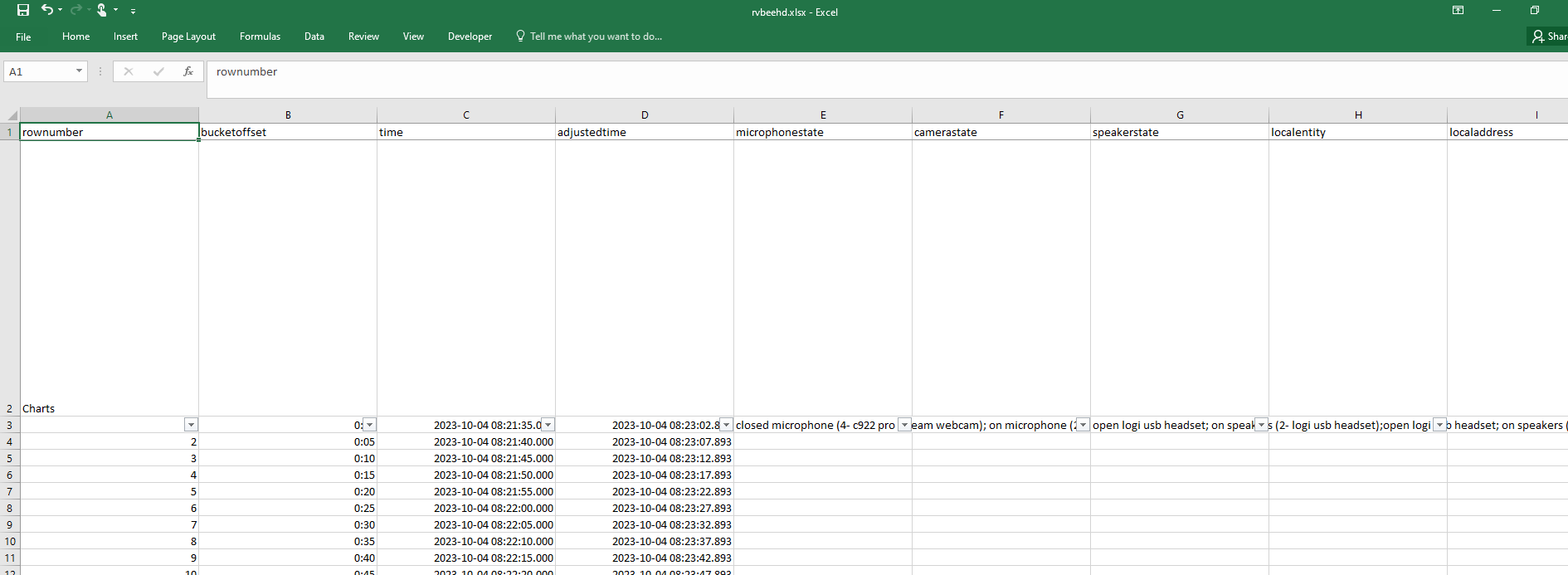


#### Session worksheets

* The main feature is displaying charts of the various SIP session parameters – particularly relating to audio and video data sent and received



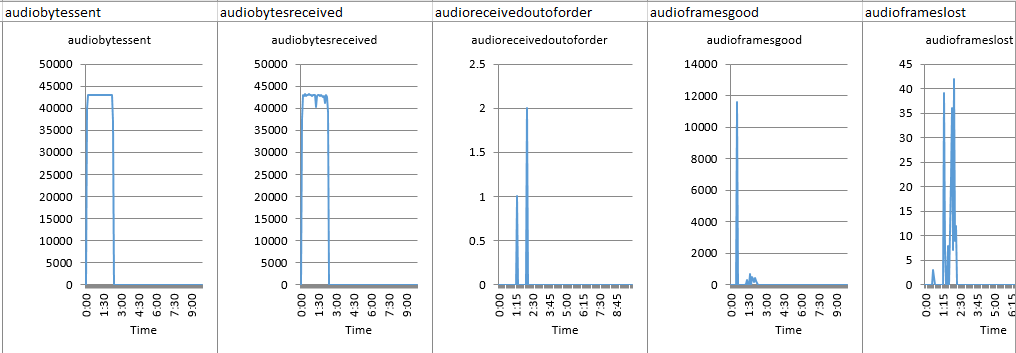
* The payload lines in Messages were scanned to identify the time periods covered by individual ATM-Teller SIP calls. Using a standard call duration of 10 minutes and a granularity of 5 seconds, the statistics were summed into ‘time buckets’ and then the array of buckets displayed in the charts.



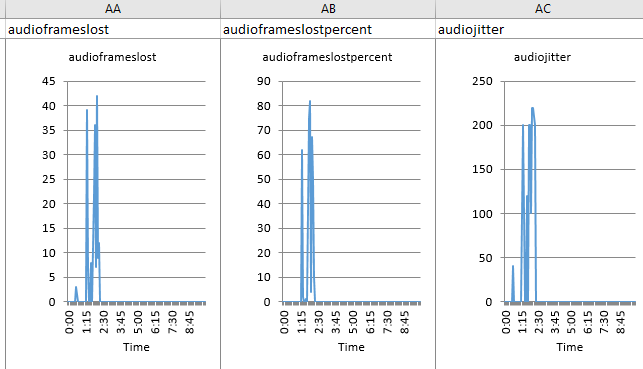
* The most important value in the charts is the performance of audio and video data sent to the remote side, or received from the remote side. In this way outages of video or audio can be easily seen.

In an ideal network, all of the audio bytes sent would be received and the same for video. It is useful to compare the charts on both sides of a call and even compare multiple tellers and terminals, to identify a pattern of network performance issues in a bank’s network.  
  
Audio and video are sent in UDP packets, which might not be prioritized in their travel in the network from source to destination machine. That means if a router is congested it might drop a UDP packet completely, or try to reroute around another congested router so it would take longer to reach the destination end. That forces the receiving side to freeze video or audio, for a very short time (choppy audio or video) or much longer (a second or two, or longer).  
  
Audio UDP packets are all of similar size and normally would be expected to receive smooth passage across the network. Video UDP packets vary quite a lot – some (‘key frames’) are very big, others smaller. As a result, expect more problems with video than with audio.

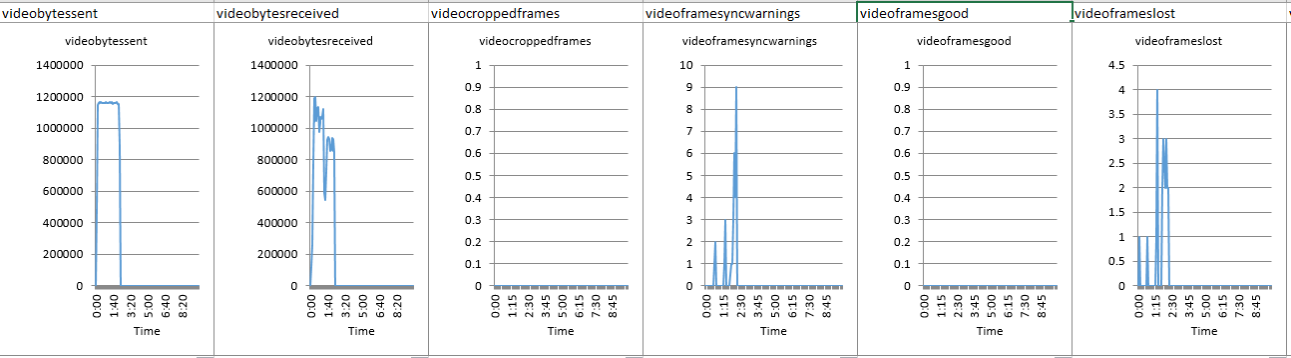
* Key charts to look at:
  + audiobytessent
  + audiobytesreceived



* + - for bytessent, notice that the number of bytes sent rises quickly at the beginning of the call, remains at a smooth high level, and tapers quickly down at the end. This indicates that sound was being transmitted to the remote side throughout the call. If there were any gaps at the beginning or in the middle of the call, that would indicate a problem or the call being placed on hold or the mic muted [*need proof of these last statements from logs*]
    - for bytesreceived, notice the rise in the chart but it is not so smooth at the top. Looking at the charts to the right there were some problems receiving the audio frames – possibly due to network congestion. If the problems are bad enough, the teller would report choppy or lost audio.
    - SET LOG LEVEL APPROPRIATELY (DEBUG) to see bytessent and bytesreceived in the BeeHD logs. To do this edit the BeeHD configuration files [Logs] section and add DEBUG to transportLogFilters:  
        
      *transportLogFilters=INFO,ERROR,EXCEPTION,WARNING,INFO,****DEBUG***
  + audioreceivedoutoforder
  + audioframeslost
  + audioframeslostpercent
  + audiojitter
    - these all indicate audio reception problems (data lost in transmission over the network)
    - jitter is a measurement of delay in msec between when a packet was sent and when it was received and processed. BeeHD can tolerate and smooth out a certain level of jitter, but beyond 120 msec or so you can expect complaints from the tellers or ATM users



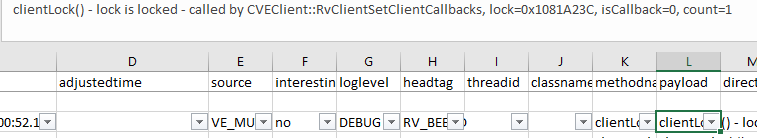
* + videobytessent
  + videobytesreceived
  + videoframesyncwarnings
  + videoframeslost
  + videoframeslostpercent
  + videopacketslost



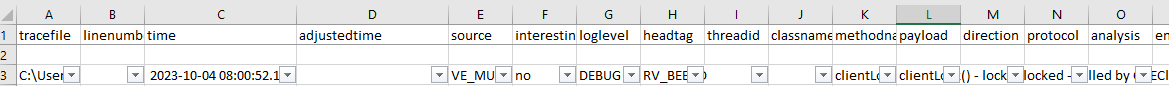
* + - similarly for video, bytessent is smooth but bytesreceived looks pretty bad in this example.

#### Message worksheets

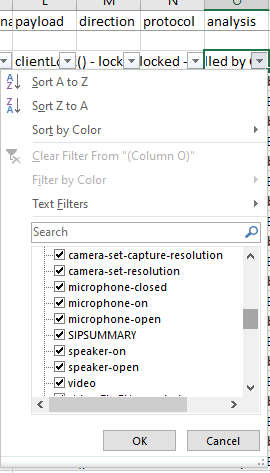
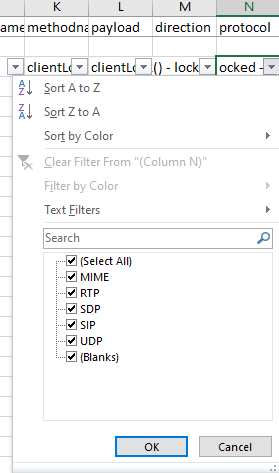
* Contain one row for every line of the input rbveehd.log file
* The text of the row is found in the ‘payload’ field



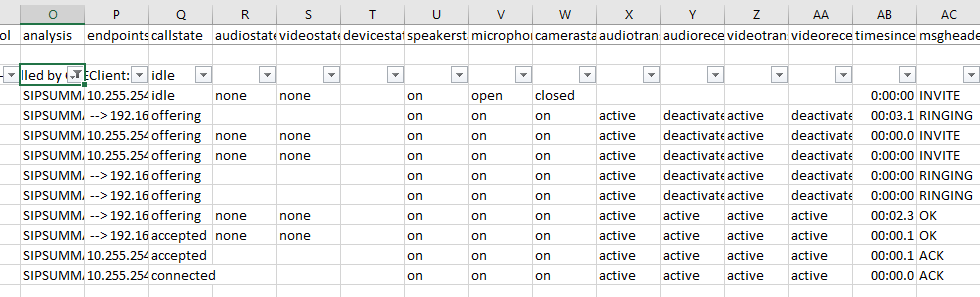
* The other columns contain an exploded view of the payload



* *The columns were split in this way as phase 1 of deciding how to analyze the log files. Phase 2 was to output a SIP session analysis in the atparser.log for each session, and Phase 3 was to present each session separately in an Excel worksheet along with performance graphs*
* Filtering the values of some key columns simplifies analysis of the log
* For example the ‘protocol’ and ‘analysis’ columns



* In ‘analysis’ select only SIPSUMMARY to see the SIP message exchanges



* Or in ‘protocol’ select only SIP and SDP to see the SIP message text. Adjust the payload column width also.

