2023

MRTC Gateway

Mizutech WebRTC-SIP Gateway –User Guide

Connecting WebRTC clients to SIP servers

Mizutech
6/6/2023

Contents

About

The Mizu WebRTC Gateway (MRTC) is a software solution to convert the WebRTC protocol family to the SIP protocol family.
The MRTC software runs as an NT service on Windows operating system and includes all modules for efficient protocol conversion between WebRTC and SIP.

Features

The most important features are listed below:

- Full stack conversion between WebRTC and SIP
- GUI based management with real-time monitoring and detailed statistics
- Multiple SIP server support for both outbound and inbound
- Convert Websocket (WS/WSS) to plain clear text SIP signaling over UDP or TCP
- Convert WebRTC media (DTLS/SRTP) to plain RTP/RTCP over UDP
- Media path negotiation with ICE, TURN, STUN
- Optimal media and codec negotiation between WebRTC and SIP
- Codec transcoding when necessary (handled automatically by default and can be forced/disabled)
- Supported codecs: G.711, G.729, G.723, GSM, iLBC, OPUS, Speex, G.722, VP8, H.264
- Flexible configuration and routing rules
- Auto TLS certificate
- Push notifications (for Web, Android and iOS)
- Intelligent NAT/firewall bypass using both UDP and TCP candidates and shortest path detection
- Secure: border security, media security, rate-limiter, DOS attack protection, topology hiding
- Encryption: TLS/DTLS/SRTP
- Reliable: HA setup with auto-failover and/or load-balancing, heartbeat monitoring and health statistics, threshold triggered actions, auto reroute, auto reconnect, auto redial
- Support for a broad range of VoIP features including voice, video and unified communication
- PBX/Class 5 features: DTMF, call transfer, call forward, mute, hold, re-invite, caller-id, call fork
- Extra features: call rerouting, voice recording, IM recording, file transfer, conference, voice-mail, presence, chat, SMS, IVR, P2P, callback, webcall, click-to-call and many others
- Compatible with all SIP server, softswitch or IP-PBX such as Asterisk, Cisco, Voipswitch, 3CX, FreePBX, Trixbox, Elastix, SER and others
- Compatible with all native WebRTC clients and WebRTC capable browsers, including Firefox, Chrome, Edge, Safari, Opera and WebViews

Requirements

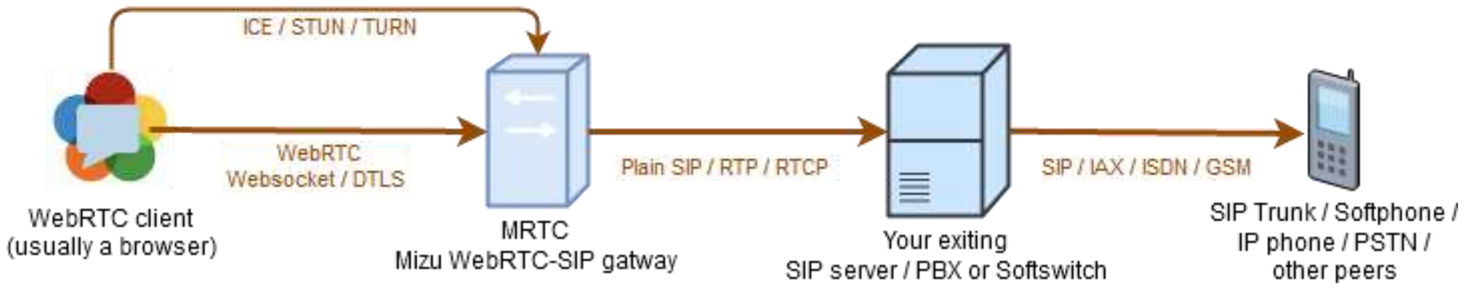
You will need the following elements to be able to use the gateway:

- The hardware requirements depends on the usage. For example for up to 10000 users and 100 simultaneous calls: a low-end server with 2 CPU core, 4 GB RAM and 60 GB disk. The service can be hosted also on a virtual server or cloud server
- Network requirements are similar to a typical SIP server (around 10 mbits for 150 simultaneous calls. Basic registrations are using minimal bandwidth: 1 mbits for 10000 users).
- OS: Windows (The MRTC gateway can be hosted on any Windows version. Recommended: latest server Standard version, 64bit. Note that your MRTC will be secure, regardless of the OS)
- SIP Server (Softswitch or IP-PBX) or SIP proxy. This is your existing SIP server(s) where you wish to add WebRTC support
- Any RFC 7118 compatible WebRTC client (such as the webphone, SIP.js, JsSIP, sipML5 and others. Usually ran from a browser but you can also use native WebRTC clients)

More details can be found here.

How It Works

The WebRTC-SIP gateway acts as a relay between the WebRTC clients (usually browsers) and your SIP server(s) (IP PBX, Softswitch, SIP proxy or other SIP capable equipment). MRTC includes all the necessary modules for optimal protocol conversion regardless of your WebRTC or SIP software and network circumstances. The gateway will transparently convert the traffic from the WebRTC clients to plain SIP/RTP. No need for any changes or re-configuration on your existing server.



More details about how it works can be found on the MRTC homepage. General technical details about WebRTC-SIP gateways can be found here.

Legal Agreement

Mizu®, MRTC, webphone, Mizu WebRTC-SIP gateway, Mizu WebRTC-SIP proxy the names associated with Mizutech products are trademarks and/or service marks of Mizutech SRL. All other trademarks are

property of their respective owners. This product may contain open source software freely downloadable from Mizutech website.

This SOFTWARE is licensed, not sold. All rights reserved for MizuTech SRL.
By installing, copying, or otherwise using this product, you acknowledge that you have read, understood, and agreed to be bound by the terms and conditions of the End User License Agreement for this product.

The software is provided "as is" without any warranty of any kind.
Mizutech makes no warranties, liability or responsibility for the accuracy of this document and makes no representations about the suitability of the information contained in this document for any purpose. In no event shall Mizutech be liable for any direct, consequential, incidental, special, punitive or other damages caused by this document or the MRTC software.

You can order and purchase a license for this software from here.
Contact us with any questions at info@mizu-voip.com.

Setup and configuration

This is the most important chapter in this Guide.

In short, you just have to do the followings:

1. Download and install MRTC
2. Configure MRTC from the MRTC Admin Configuration Wizard:
   Just follow the instructions (set Bind/Local IP, Domain name, SIP server)
3. Start the service and make sure that the executables are enabled on your firewall
4. Configure your WebRTC clients (SIP server, Websocket URL, ICE/STUN/TURN) after the instructions at Help menu -> Client Configuration

Details below:

Install

The MRTC gateway has a standard easy to use installer.
The gateway shall be installed on a Windows machine (server or PC) somewhere near your softswitch OR near your customers OR in the path between (to minimize media delays).

1. Make sure that your system meets the minimum requirements.

2. Download the MRTC installer from here
3. Double-click to start the install process and follow the instructions (requires Administrator rights)
4. Once the install completes, it should automatically start the Admin client with its Configuration Wizard. This is discussed in the next chapter.

Configuration Wizard

For the basic server configuration you should walk through the detailed **configuration wizard** which is started automatically at first usage or accessible from the "Config" menu.

Don't change any setting that you don't fully understand, just click on the "Next" button in this case. Most of the settings are self-explanatory with a short description near them and/or a hint if you hold the mouse over a control.

Pay **special attention to the "Network" and "SIP server"** pages:

- on the Network page enter the correct IP and NAT configuration ("Offer services for" or "Allow LAN peers")
- make sure that the ports used by the WebRTC-SIP proxy (SIP port, Access port, Secure port) are not used by some other application such as a local web server (in this case either change the MRTC ports or the third party app port or bind them to separate IP address)
- you need a domain for TLS certificate, otherwise it won't work in modern browsers (both the domain and the TLS certificate can be auto generated or alternatively you should assign a sub(domain) for the gateway IP and you can also set your own certificate)
- on the SIP server page enter your PBX/Softswitch details (later you can add more SIP servers if needed)

Click "Apply" on the last page to save the configuration. Check the displayed messages and instructions. If you select the "(Re)Start the server" checkbox, then the gateway service should start automatically. Otherwise start it from the "Control" menu or use the Windows Service manager and start the "mserver" service.

When you start the service for the first time, it is a good idea to open the "Analyze" form to check for any potential issues.
Once the service is started, you can connect with WebRTC clients and make calls.

Later you can add more servers or modify advanced configuration options after your needs.

Note: for WebRTC to work in modern browsers, you need to host your webpage on HTTPS (secure HTTP with valid SSL certificate) and you must use WSS (secure Websocket) for the WebRTC proxy connection.

This means that you must specify a (sub)domain name and enable auto-ssl or set your own certificate for the gateway.

SIP Server Configuration

A default/first SIP server record is created for you by the Configuration Wizard from the details you provide on the “SIP Server” page.
Use the “Add Upper SIP server” form (available from “Config” menu -> Quick add -> Users) **in case if you have more SIP servers** where the traffic from the clients needs to be sent (WebRTC -> SIP) or accepted from (SIP -> WebRTC). Also don’t forget to manage the routing in this case.
The most important configuration for a SIP server is its network address (IP and/or domain name). Also you must specify the port number if your server is not using the default SIP port (UDP 5060). This can be done by just appending it after the IP after a colon (such as 11.22.33.44:5678) or as the “port” field in the user table.
Note: if you enter the :port after the IP then the port part will disappear, but the port field will be set accordingly.
You **don’t need to change any setting on your existing SIP server**/PBX/Softswitch as you will see all the traffic coming from the WebRTC clients as normal SIP traffic, usually with the SIP standard digest authentication. The only notable difference is that all traffic from WebRTC clients will come from the Gateway IP, so make sure to white-list this IP on your server or release any rate-limiter restrictions, but do not set IP based authentication for the Gateway IP! (The registrations and calls from WebRTC clients will do proper SIP digest authentication, so you will need to authenticate them as endusers/extensions and not as a SIP trunk with IP authentication, unless you do this intentionally, to unconditionally accept any traffic from WebRTC).
More details about configuration can be found here and here.

WebRTC Client Configuration

In short: just **use the “Client configuration” from the “Help” menu** for the exact details.

Details:
Once you finished with the configuration wizard and your gateway is running, you can connect with your favorite WebRTC client and make calls.
Parameters:
“Gateway address” means your MRTC gateway IP or domain name (if you have assigned a domain)
“Mainport” means the gateway websocket port (use the secure port if you enabled SSL)
If you have SSL enabled, use WSS for websocket. Otherwise use WS.

- SIP domain/proxy/realm: your SIP server domain name or IP (for example 11.22.33.44:5060)
- WebSocket Server URL: the gateway wss://address:mainport/mfstwebsock (for example: wss://domain.com:443/mfstwebsock)
- TURN server address: the gateway address and the main unsecure port (for example domain.com:80)
- STUN server address: the gateway address and the mainaportudp port (for example domain.com:8090)

To find out how to configure your WebRTC client go to the "Help" menu and select the "Client configuration" item. Use the details listed in the "#### WebRTC" section. This will list all you need to know to connect with various WebRTC clients.

Note: if you haven't set a domain and SSL certificate for the gateway or your webpage is not on HTTPS (secure HTTP with valid SSL certificate) then WebRTC will not work from modern browsers.

For the first test call use two enduser accounts (extensions) valid on your SIP server and make calls between them. Use the first account in your WebRTC client (such as webphone, sipml5 or jssip) and the second account in a SIP client (such as MizuPhone or X-Lite). Register with both and call from WebRTC to SIP then from SIP to WebRTC.
Please note that you can also make calls between WebRTC and WebRTC and also between SIP to SIP endpoints (although this latter usually doesn't make much sense, because there are no reasons why you might connect with SIP clients via the gateway, since your SIP server can handle these anyway).
During a call, you can open the "Current calls" form in the MizuManage to see the details.
After the call you can see the CDR (call detail report) by opening the "CDR" form in the MizuManage. If there are no CDRs (call detail records), it means that the call has not reached the server (wrong network settings on server or client side).

Troubleshooting

If you can't connect/register or make calls, check the followings:
- Make sure that the service is actually started (check the "mserver" service status in Windows Service manager)
- Make sure that the application is enabled on your firewall
- Have a look at the "Dashboard" form
- Open the "Analyze" form and click on the "Analyze" button to detect potential issues
- Check any important errors or warnings on the "Logs" form

- Make sure that you have basic connectivity to the important ports such as 80 and 443. Use telnet gatewayip port command from a PC from where you are trying to connect with WebRTC
- Check if TLS/certificate is installed and working correctly (MManage -> Config menu -> Utilities -> Tests -> TLS certificate test)
- Install a SIP softphone (such as X-Lite or MizuPhone) on the same server where you are running your Gateway and make sure that you can make call to your SIP server from there.
- If you are making call to a user/extension, make sure that the called user is registered when you call it
- If you don't hear any voice you might change the RTP routing for the user(s) to "always route RTP" from MizuManage -> Users and devices -> Edit tab
- If the call setup time is too long, check here
- Are you running your WebRTC or SIP endpoint on the gateway PC? Don't do this. Most WebRTC stacks has various bugs when they are run from the same IP where they connect to.
- In case of call failure you can check the disconnect reason from the "CDR" form
- Verify the log files. You can find the logs in the server app folder (near the mserver.exe) -> "logs" subfolder. You can also open the folder from MRTCAdmin -> Files Menu -> Folders -> Server logs.
- Open the last log file ("log_xxx.dat") with a fast text viewer (For example F3 from Total Commander). To find application errors, open the last log file in the server app directory and search for "ERROR" and "WARNING". To find a call, search for "INVITE sip:callednumber".
- You can modify the trace details with the "loglevel" configuration option: from 1 to 5.
- Contact Mizutech support if the problem persists and you have or wish to buy a license

Management and Monitoring

The most important settings can be configured from Config menu -> Configuration Wizard as already discussed in the previous chapter.
Once the WebRTC-SIP gateway is installed and configured correctly, there is nothing much to manage on the Gateway itself as it has full auto-management capability (such as auto deleting log files, auto fine-tune to your hardware, automatic user management and others).
This means that this whole chapter is completely optional and you might not need to know about these details at all to run a fully working WebRTC-SIP proxy.

MizuManage

All administration and monitoring tasks can be done from the MRTCAdmin client (MizuManage / MManage / MizuManagement), which is included in the WebRTC gateway install package.
Using the Gateway, you will still need to use your Softswitch/PBX to manage your Users (authentication, authorization, and accounting), Routing, Billing and any other aspects of your network.

Regardless of this, the WebRTC-SIP proxy also provides you complete monitoring and GUI based configuration managements, so you can easily change settings or find connectivity issues if that is WebRTC-SIP conversion related.

Always check the status bar (the bottom text display). MRTCAdmin rarely displays popups and the success/error state of various operations are displayed on the status bar instead.

Filters:
Nearly all forms in MRTCAdmin can be filtered with the followings:

- Quick filter: found in the top-left side in MizuManage. For example type “44*” in the quick filter box then open the “CDR” form and click “Load”. You should be able to see all calls to 44….. numbers. Or enter “test” and open the “Users and devices” form. Click on the load button to see accounts containing the “test” word (in name, username, address, etc)
- Direction filtering: accessible by double clicking on the space below the quick filter or from the Fields menu -> Filter -> Set direction filter. When you are doing operation which needs more precision (eg. billing), always use the Set Directions form and not the quick filter.
- Date-Time filter: found in the top-left side in the MizuManage. Useful to restrict statistics, reports and CDR listing intervals.

Menu:

- The most important item is the Configuration wizard (from Config menu). Use this to (re)configure your gateway most important settings.
- From the “Control” menu you can Stop/Start/Restart your server (you can do so also from windows Services management console)
- To export data from the application, use File menu -> “Save As” (when the desired data is displayed)
- From the “Fields” menu you can manipulate the selected dataset (grids, etc).

User Management

The MRTC gateway works transparently so you will have to continue to manage the users on your SIP server as you did it before. On the gateway you might configure only SIP server and Traffic sender user types to point to your SIP servers (if you have more than one SIP server, otherwise a single SIP server can be also configured from the Configuration Wizard).

All user records can be managed from the “Users and devices” form (below the “Access” node in the tree-view).

Users can represent real people, endpoints, extensions, devices, gateways, proxies or servers.

The most important user types are the followings:

- SIP server for outbound routing. Your SIP server(s) should be added here.
- Traffic senders for inbound routing. Your SIP server(s) should be added also here.
- Endusers: these are managed by the Gateway automatically.
- Admin users: you should add yourself (and/or the administrator) here as it is used for various purposes such as sending alerts and reports.

The users are stored in the database tb_users table and they have a lot's of properties which can be managed from the "Users and devices" form or (for advanced users only) changing database record fields (select a user the go to Fields menu -> Show fields).

You don't need to manage endusers/extensions on the Gateway! In the MRTCAdmin you might see your endusers (extensions) listed on the "Users and devices" form, however that is only for monitoring purpose and for the gateway internals, auto created and managed by the Gateway. Except your extension usernames, no authentication credentials (password) or any sensitive information (such as billing) are stored on the Gateway. Use your Softswitch/PBX for user management.

Listing users

Open the "**Users and Devices**" form (below the "Access" section) and (double) click on a user type to have a listing.
You can apply various filtering using the user "Type" checkbox-list, the dropdown-list on the top of the form or the already discussed direction filter or quick filter.
Your SIP servers can be found as "SIP Server" and "Traffic sender" accounts.
You can easily search for users using the "quick filter" box. For example to list all outbound routes whose username or name contains the "carrier" word, select the "SIP Server" and type "carrier" in the quick filter. The quick filter will also search in other fields such as IP, name, email and others.
Other listing options are available also from right click on the "Users and devices" node in the main tree-view or right click on user type node from within the "Users and devices" form.

Creating users

The best way to create new users is to clone an existing working account with the same user type.
For this, launch the "**Users and Devices**" form, select a user type, and click on the "Load" button. Then select any user entry and click on the "New User" button. Alternatively right click on the "New" button for other options to add user records.
Endusers are created automatically by the server at first time when an extension will try to register or call. These enduser records are only for internal management and statistics reasons. This are the same users as the extensions on your PBX, but the Gateway might have information only about their username and display number (the gateway does not have to know the password since the

authentications are forwarded to your SIP server). You don't need to manage these endusers on the gateway (Keep using your PBX to manage your users).
The username can be also used as a real phone number; will be used also as the caller-id if not specified otherwise. Endusers can make voice or video calls, presence and chat directly between them without the need to be forwarded to your SIP server. (Alternatively you can disable direct routing between endusers by setting the alloweuserusercalls and fs_directroutrfstofs global config values to 0).
Traffic sender users are used for receiving traffic from your SIP servers (PBX, Softswitch, SIP proxy or other SIP device). The authorization type is usually set to "Auth ip must match" and you have to enter a correct "Auth Ip" (IP address based authentication).
SIP servers: For outbound traffic you need one (or more) SIP Server user. The most important parameter here is the "IP" where the VoIP calls will be sent. Then you will have to configure these also on the Routing form.
To be able to send and receive traffic to/from another SIP server or carrier you will have to add it as both a "traffic sender" and "sip server" user.

Add SIP Server

Your upper SIP server(s) can be added multiple ways:

1. from the Configuration wizard you can add your default/first SIP server on the "SIP server" page
2. from the "Add Upper SIP Server" form (from Config menu -> Quick add -> Users)
3. by SQL query or API call (see the Integration chapter)
4. from the "Users and devices" form (This is discussed here below)

A default/first SIP server can be added during the configuration wizard. (You can also modify it from right click on "Configurations" node in the tree-view and select the "Upper server" from the popup menu).
The WebRTC-SIP proxy can handle multiple upper servers in case you have more SIP servers where you wish to accept WebRTC traffic.
To add a SIP server open the "Add Upper SIP server" from Config menu -> Quick add -> Users. (Alternatively right click on the "Users and devices" node in the tree-view and select "Add Upper SIP Server").
Pay attention to the following fields: IP, domain, port (you can also set the proxy field if needed).
Once you added your server, make sure to modify the routing after your needs (if you have more than one server, then the routing needs to be able to decide somehow which traffic to send to which server. You can create various rules for this on the "Routing" form)

When you add a SIP server, this is what happens:

1. a Traffic sender account is created which will handle inbound traffic coming from your SIP server (SIP -> WebRTC)
2. a SIP server account is created which will handle outbound traffic to your SIP server (WebRTC -> SIP)
3. a default routing entry is created for the above SIP server

Actually, you can "add a SIP server" also by just adding the above accounts manually from "Users and devices".

If the WebRTC clients have to register to different SIP servers, set the "routingforregister" global config option to 1 (from the "Configurations" form). Otherwise set to 0. If the "routingforregister" is set to 0, then it will forward all registrations to your default SIP server which you have created during the configuration wizard.

More details about multiple SIP servers can be found here.

Add User

There is **no need** to add endusers, extensions or WebRTC clients manually.
Just continue to manage your users as before: on your SIP server.
You will be able to see the users on the "Users and devices" form -> Endusers, however these are managed automatically by the Gateway and there is no need to change any settings here except if you have some extra requirement such as setting call forwarding, call recording or some other per user setting. The first time a WebRTC client will register or make call, a user entry is automatically created by the Gateway. This is mostly for internal usage only, but it also allows you to make per user changes if necessary. No password of these users are stored on the WebRTC gateway (The users are created with "Username" based authentication only, however the gateway will also check other parameters such as same session / same address for further account restriction. If you wish, you can create additional users of your own for various other purposes.

For special purposes you might add users also manually. There are multiple ways to do this:

- click on the "New" button from "Users and Devices"
- or right click on the "New" button
- or right click on the "Users and devices" tree-view node.
- or from "Config" menu -> "Users"

Outbound Routing

Outbound routing rules are required for the gateway to properly route the calls to your SIP server(s) (for WebRTC -> SIP calls).

Calls between users are handled automatically as this will bypass the routing table and it is routed directly unless if you disable it. However for outbound routing you need to add a "SIP server" user first and then add it to your routing rules.

Once you add your upper server as a “SIP server”, open the “**Routing**” **form**.
In the left side you have to define your pattern which will restrict the condition when the actual route entries can be used. If all fields are empty and the time definition is set to “All times” then all patterns will match. You can make restrictions if you make specifications here (caller, called prefix, time restriction, etc). Make sure that you increase the priority for the pattern (to be higher than the “general” pattern where you have not made any restrictions).
On the right side you will have to add one or more sip proxy users. If you set more than one route with equal priority, then you have load balancing, LCR or BRS (depending on the “brs_lcr” global config option); otherwise the traffic will be routed after the prioritizations (will flow to the lower priority servers only if you have reached the maximum port limitations or because automatic failover).

If you have only one server then just create one pattern (left side) enabled for all times and no any filter set, then assign your SIP server user entry to this pattern (right side).

For more details please read the routing guide.

Client driven routing

The outbound routing can be influenced also from the client side.
In this case the client will suggest the target upper SIP server with the INVITE target URI (so you will configure the gateway as a SIP proxy in your app and will set the SIP domain to the target upper SIP server) or by the X-Sy.Uppersrv (address), X-Sy.Uppersrvd (domain), X-Sy.Upperproxy (proxy) extra SIP headers.
This is considered by the gateway only if *allowupperserverselection* global configuration is set to *1*. It can be set also from the Config Wizard -> “SIP Server” page -> “Allow client driven upper server selection” checkbox.
In this case you should still create the SIP Server user entries and add them to the Routing (usually with equal priority, so the target server will be selected based on your app preferences)

Dial plans

You can manipulate number format, SIP headers or the Caller-ID from the following settings:

- Users and devices form: caller-id, username, other numbers (DID match), tech prefix
- Routing form: you can only specify routing direction here without number changes
- Rules form: this is a powerful module which you can use to change almost anything (including caller-id, called number format and many others)
- Global configuration: a few global configuration options might also affect the dial plan. Right click on the “Configuration” node and select “Number rewrite” for the details.
- Prefix rules and the dial plan form: use the “Rules” form instead when possible.

You can find more details in the VoIP Admin Guide below the Routing section.

Inbound Routing

If you would like to accept traffic from your servers (SIP -> WebRTC calls), you need to have one or more “**Traffic sender**” user account created. Usually you can use IP based authentication. For this, add the peer IP to the “**Auth IP**” field.
For each incoming call, the server will first check if the called party is a local user. If not, then the call is routed according to the rules which are set by the “Routing” form.

One “Traffic sender” user entry can handle incoming traffic with IP authentication from multiple IP addresses. Use the “…” button near the “Auth IP List” to add more IP addresses.

Usually there is no need to setup any routing rule for inbound calls (as these will be calls to local WebRTC users which are found automatically).

Traffic between endusers/extensions is handled automatically.

CDR

A CDR (call details report) is generated for each call on your gateway.
You can access these records by using the **“CDR**” form under the “Monitoring” node in the tree-view.
By default only the most important fields are listed (date-time, connect time, call duration, etc). You can see more details if you check the “All fields” checkbox.
To quickly list the CDRs that belong to a user, open the “Users and devices” form. Find the user record, then right-click on it and select “Set Direction”. Than go back to the CDR form and click on the “Load/Reload” button.
You can easily filter calls with a specific prefix by typing the prefix in the quick search box following with an asterisk and hit enter. For example searching for 44* will list all CDR where the called number begins with 44.
You can also see various statistics based on these CDR by using the “Statistics” or the “Disc. Reasons” forms.

Statistics and Monitoring

The Mizu WebRTC-SIP proxy provides endless possibilities for monitoring both real-time and statistics history. Some of the most important tools are the followings:
Dashboard: a summary of the most important parameters and a start point for management (Note: You can access various statistics by just clicking on the Dashboard items. For example click to “CCalls” will show the current calls)
By using the “**Analyze**” form you can have a quick overview about the system and warnings for malfunctions or configuration suggestions

Registrations: Monitoring -> "Registrar" form and "Users and devices" form
List the active sessions: Monitoring -> "Current Calls" form
Disconnect reasons: Monitoring -> Disc. Reasons will show statistics about call disconnect codes.
Devices registrations: Monitoring -> Registrar will show details about registrations
Statistics by SIP server: Monitoring -> Statistics -> Group By: called servers (useful if you have multiple server, softswitch or PBX)
Statistics by users: Monitoring -> Statistics -> Group By: caller
Statistics by day: Monitoring -> Statistics -> Group By: day
Logs: the most important logs can be listed from the "Logs" form. Detailed logs can be found in log files accessible from Files menu -> Folders -> Server Logs directory
Other more **advanced statistics** can be generated by using the Statistics form and using different fields/options/grouping/directions.
All statistics can be filtered by the "set direction" form or the "quick filer" edit-box and by a time interval selection.
Automatic reports: The MRTC Gateway can send daily reports for administrators or email/sms alerts on malfunctions. For this you have to setup an "Admin" user with a valid email address (From the "Config Wizard" or from "Users and Devices"). Then set the following user fields to 1 (after your needs): "sendemailalert", "senddailyemal", "sendmonthlyemail", "sendsmsreport", "sendsmsalert".

Start/Stop

You can start/stop the WebRTC-SIP gateway service from:

- the "Control" menu in MRTCAdmin
- or by the Windows Service manager locate the "mserver" entry, right click and select "Start" or "Stop"

Note: the first start might take a bit longer (around one minute), otherwise the start-up time should be below 10 seconds.

Backup/Restore

A backup for MRTC can be done with simple file copy (xcopy) and just copy back the files for restore.
You can back-up all your data and settings by just copying the mserver.sdf file from the app folder
Alternatively, you might choose to copy the entire app directory, excluding logs (however these files can be easily recreated by a reinstall, so you really need only the mserver.sdf).
By default the Gateway will automatically create nightly backups and auto-delete old backups (Search for "backup" in the global configuration to change this).

Additional Settings

The below discussed settings have to be changed only if you have some special requirements, otherwise the default settings are optimal values for a successful WebRTC-SIP protocol conversion.

Through this guide we often refer to various configuration keys also called as “global configuration” or “global config”. All these can be managed from the “Configuration” form (below the “Other” node in the tree-view).

Settings

The gateway has a long list of configurable settings which you might adjust to turn on/off features or to adjust the WebRTC-SIP proxy behavior after your needs.
The settings can be categorized in the following way:

- global configuration (editable from the “Configuration Wizard” and from the “Configuration” form)
- per user configuration (editable from “Users and devices”)
- other configurations (editable form various forms such as “Routing”, “Rules” and others)

Global Configuration

You can quickly change any global configuration from the “Configuration” form (below the “Other” node from the main tree-view). Just search for your keyword to find the related setting(s).
Most of the changes can be applied instantly (click on the “Apply Now” button), however some changes (such as local/bind ip/port reconfiguration) require a restart.
Some settings groups can be accessed by right clicking on the “Configuration” form.
SBC related options can be listed by right click on the “Configuration” node and select the “Gateway” from the popup. Important gateway related settings are the following:
autocreatereguser: 0=no,1=when fwd authenticated ok register, 2=always (when we receive the register)
fwdregistrations: 0=no,1=only from alternate port, 2=always
fwdregistrations_address/fwdregistrations_domain/fwdregistrations_ip/fwdregistrations_port: the address of your SIP server (the upper server)
forwardauthentifications: 0=no (default),1=yes,2=yes with username as callername, 3=yes with phonenumber as callername, 4=yes with username as authname too, 5=yes with phonenumber as authname too, 6=replace authorization with username but leave the A number intact, 7=replace authorization with sipphonenumber
forwardauthpassword: 0=fwd from ep, 1=means 2 if from webrtc for recent users, 2=answer from local always,3=answer from local also for register, 4=answer from remapped upperusername/upperpassword

fs_restrictsiptolocal: calls from SIP will be allowed only to webrtc. 0=no,1=yes if mizu gateway (def),2=yes if gw, 3=yes always (2 and 3 should not be used)
fs_checkdomainmismatch: useful if you have multiple servers with different users and wish to avoid mismatch routing. 0=no,1=yes (def if fs_ismzgateway > 1),2=always

More WebRTC related options are listed here.

Clean Install

By default when you reinstall the MRTC service, it will keep your old settings and data, thus you can easily upgrade to new versions.
If you wish to begin with a clean state, delete (backup first!) the old mserver.sdf file before launching the installer, thus you will have a clean state with all your previous settings and data erased.

Domain name

Certain browsers require HTTPS for WebRTC to work (both for your website and for Websocket/DTLS) and to be able to setup a valid SSL certificate for your WebRTC-SIP proxy, first you need to assign a domain name to its IP address.
The domain name is also useful if later you wish to change your Gateway IP address or location (in this case you will just need to redirect the domain to the new IP and everything will continue to work as previously).
If you don’t already have a domain, you can purchase it easily from any domain registrar company such as GoDaddy.
If you already have a domain (for example created for your website), then you can easily add a sub-domain as most domain registrars provides an online user interface where you can add sub-domains for free.
A sub-domain is perfectly fine for the WebRTC-SIP gateway, for example rtc.yourdomain.com. The sub-domain must have an A record set to your Gateway IP (ipv4 address).
Also the WebRTC gateway can provide you a free domain name with a single mouse click: on the Configuration Wizard click on the “get” link near the domain name to obtain it (provided by Mizutech auto-dns service). Please note that it might take some time until the domain will be assigned for your server (sometimes instantly, but sometimes it might take 2 or more hours). However, in production it is a good idea to assign your own domain name (which is owned and controlled by you).
You can use the nslookup or the ping tools from your command line to see if the domain assignment is correct.

For example:

```
nslookup rtc.yourdomain.com
ping rtc.yourdomain.com
```

See if the response lists your MRTC Gateway IP. If you have recently assigned the domain, then you might have to wait a few hours for the dns cache propagation.
Your OS might also cache (incorrect/old) dns record. Use the ipconfig /flushdns to clear the cache and recheck again with one of the above commands (ping or nslookup).

SSL certificate

If you are using a web based WebRTC client (not a native library) then most probably you will need an SSL certificate. This is because modern browsers will allow WebRTC only if you are using secure websocket (WSS). Self-signed SSL certificates will silently fail, so you will need a valid certificate. Also your webpage from where you launch your WebRTC client need to be hosted on HTTPS (or test from local file).

"SSL certificate" is a commonly used term but in fact it really means signed RSA keypairs used for TLS encryption.

The Mizu WebRTC gateway can acquire and maintain a **free Let's Encrypt certificate automatically** which is just perfect for WebRTC.
To enable this, make sure to select the "Auto get SSL/TLS certificate" checkbox on the "Network" page of the Configuration Wizard (or set the "autotlscert" global config to 1).

The following are the prerequisites for the auto-ssl to work:
1. the gateway must have internet access
2. you must have a valid domain name assigned to your gateway IP (discussed in the previous point)
3. ports 80 and 443 must be available for the Gateway (not used by other apps such as a Web server)
4. if the gateway is behind NAT, make sure to forward ports 80 and 443 correctly

Note: starting from MRTC v.2.4 (2018 Apr.) there is no need to forward ports 80 and 443 anymore if you acquired a webvoipphone.com subdomain since the new version is capable to utilize already prepared wildcard certificate for this domain.

If you prefer, you can also use **your own SSL certificate** if you already have one and it is assigned for the (sub)domain which you have set for the Gateway. For this you just have to copy the following files

in the gateway directory (by default at C:\Program Files (x86)\MRTC\ or accessible from MRTCAdmin -> File menu -> Folders -> Server App directory):
key.pem: the key file (your private key which you have got when generated the CSR - keyfile.key)
cert.pem: the certificate file (your certificate as received from the CA –yourdomaincertificate.crt)
root.pem: the chain: CA intermediary + root certificate. intermediate.crt + root.crt. The root.pem is not so important and eventually might not be used at all in some circumstances or it can be loaded from the OS trust chain.

In case if your certificate has a password, you can configure it with the sslcertpwd global config (this usually applies for PKCS certificates: pfx or p12 files).
In case if your certificate is another format (such as p12), you can easily convert to pem with this tool.
New versions (since July 2020) is capable to automatically use cert.crt, ca_bundle.crt or certificate.crt or convert from cert.p12, cert.pfx, certificate.p12, certificate.pfx.

In case if you are not familiar with obtaining TLS certificates and can't use the built-in Let's Encrypt, them the sslforfee website offers a free and easy way to create certificates.

You can use the Gateway also **without an SSL** certificate. In this case use unsecure websocket (ws:\\IP:port) from any native WebRTC client or some browsers which doesn't require secure websocket (It will not work from modern browsers).

SSL certificate on local LAN:
Since it is impossible to acquire a valid certificate for a local domain assigned to a private IP address, you can do the following workaround in case if you wish to run MRTC on a private network:
1. Create a (sub)domain and assign it to any existing web server IP and request a certificate for it (since you can easily request TLS certificates only for public webservers)
2. Copy the certificate files to the MRTC app directory (.pem files as described above)
3. Redirect the above (sub)domain to your gateway private IP address (just rewrite it's A record settings)

Note: starting from MRTC v.2.4 (2018 Apr.) you can just get a webvoipphone.com subdomain for your private IP since the new version is capable to utilize already prepared wildcard certificate for this domain.

Change IP address

If you migrate the WebRTC-SIP Gateway to another box or your IP changed, then you will need to set the new IP.
There are 2 ways to easily change the Gateway IP:

1. going through the Configuration Wizard and make changes on the "Network" page
2. or right click on the "Configuration" node in the main tree-view (below the "Other" node), select "Network -Basic", search for the old IP and rewrite to the new one (localip, bindip, domain, etc)

Restart your gateway once you changed the IP because this can't be applied at runtime.
Note: the gateway is also capable to auto-detect it's IP address configuration if you don't explicitly set it's bindip and localip.

Ports

Since VoIP servers are using a wide range of ports, it is recommended to host it on a public server (with a public IP), unless you plan to use it only from an internal network. In addition, it is recommended to turn off any external or third party port based firewall (you can still use the Windows built-in firewall where it is easy to set exceptions for applications and not for ports).

This section is important if you are hosting the WebRTC-SIP proxy behind a NAT or firewall for some reason and the gateway is used from the external network (such as the public internet).

The MRTC listen on several ports to offer its services: SIP signaling, Websocket, RTP/SRTP ports range, RTCP/SRTPC port range, TURN, STUN and others.
It is recommended to keep the default ports since the defaults are optimized for maximum connectivity and/or are conform to standards.
For example it is highly recommended to keep the default TCP port 80 for Websocket, TURN and API and use TCP 443 for the secure version of these. A lot of corporate networks allows traffic only on port 80 and 443 and if your gateway is using other ports the clients from behind more restrictive NAT's and firewalls will not be able to connect with their WebRTC client. If you are behind NAT, it is recommended to set one-to-one port mapping (internal listening ports mapped to the same external ports on your NAT device).

If these ports are already used by some software running on the same host, then we recommend to assign a secondary IP for your server and bind the gateway to this new IP (set the bindip), then reconfigure the other software's to use only the old IP (bind to the old IP). For example IIS can be bound to an IP with the following command:
`netsh http add iplisten ipaddress=IPADDRESS` (replace IPADDRESS with the IP to bind)

If you are hosting the WebRTC-SIP Gateway on the same box where your softswitch is running, then assign a separate IP (bindip) to the Gateway if possible. If this is not possible, then make sure to change the ports on the gateway to avoid conflicts (if your SIP server is listening on 5060 on the same host, then set a different "localport" for the Gateway).

The exact port numbers used by the SBC service can be listed from MManage -> Config menu -> Network - > Active Ports.

Here is the default **list of ports** used by the gateway and their global config options which you can change from the "Configuration" form if needed.
Mandatory ports:

- localport UDP and TCP (main SIP signaling port. Default is 5060)
- mainaport UDP and TCP (main server port for various purposes websocket, TURN, API and others. Default is 80.)
- mainaportudp UDP (used for STUN, UDP RTP relay and others. default to 8090)
- sslportmain and cfg_sslport TCP (443 by default for SSL, HTTPS, WSS)
- MinRTP-MaxRTP range UDP (for RTP and RTCP; you might enable also TCP for TCP ICE candidates)
- fs_minrtp- fs_maxrtp range UDP (SRTP and SRTCP for WebRTC and RTP for extra PBX)
- tcpcandidatesrvport (TCP relay port, 10080 by default)

Optional ports (good to allow also these):

- remote desktop TCP 3389 or 3386 (for easy server administration)
- TCP 1433 and/or 2223 (for remote management)
- adminport TCP (for remote CLI access from MManage server console. Default is 9885)
- monitorport TCP (to easy access logs from remote MManage. Default is 9889)
- forwarderport (TCP forwarder for various internal services, 11080 by default)
- localport+1 TCP (for secure SIP sips. Default is 5061 -if you enable TLS for SIP which is known as SIPS)
- ftpserverport TCP (for remote file access. Default is 9710)
- UDP: 44444 ("voice-here" functionality in MManage)
- TCP: 9886, 9889 (optional ports for admin console and logs)

Make sure to enable these ports if you are using a port based firewall.
Make sure that the ports used by the gateway are not used also by some other application such as a local web server (in this case either change the MRTC ports or the third party app port or bind them to separate IP address).

In short, you should enable/forward the following ports on your firewall/NAT:

UDP: 80, 5060, 8080, 8090, rtp port range defined by MinRTP-MaxRTP and fs_minrtp- fs_maxrtp
TCP: 80, 443, 2223, 3386, 5060, 5061, 9710, 9885-9890, 10080, 11080, 44444

The above mentioned RTP port range should be 4x the number of simultaneous calls. For example UDP ports 30000 – 40000 will enable 2500 simultaneous calls.

The port forwarding can be quickly tested from MManage -> Config menu -> Utilities -> Tests -> Check port access.

Firewall

If you are using a firewall, the MRTC application and/or ports must be configured to be enabled.

Usually most firewalls allows outbound traffic by default, thus you will have to configure only to enable also inbound connections.
Since VoIP servers usually requires a bunch of open ports (usually 4 ports per call: inbound/outbound (S)RTP + (S)RTCP) an application level firewall is more suitable then a port based firewall.
Otherwise MRTC itself doesn't require any firewall. Firewall is useful only to filter other unnecessary services to make your OS more secure as MRTC has its own built-in strong security filters, including built-in dynamic firewall to filter out unwanted traffic such as DoS attacks.

Firewall configuration:

- If you are using a **port based firewall**, make sure that all the required ports are enabled.
  You will need to enable the following ports:
    - UDP: 80, 5060, 8080, 8090, rtp port range defined by MinRTP-MaxRTP and fs_minrtp- fs_maxrtp (values from the MRTC Admin -> Configurations form)
    - TCP: 80, 443, 2223, 3386, 5060, 5061, 9710, 9885-9890, 10080, 11080, 18080, 12080, 44444

    For more details, see the ports chapter above.
- In case if you SIP server is inside your network and you need to allow only the WebRTC clients on your firewall or NAT, then it is enough to allow and forward the following ports (these are the default ports if you haven't reconfigured them):
    - TCP: 80, 443, 10080, 11080
    - UDP: 80, 8080, 8090
    - UDP: rtp port range defined by MinRTP-MaxRTP and fs_minrtp- fs_maxrtp (values from the MRTC Admin -> Configurations form. You can reconfigure them to be in the same range and the port ranges should cover 4x the maximum number of simultaneous calls for in/out RTP+RTCP)
- The exact port numbers used by your MRTC service can be listed from MManage -> Config menu -> Network - > Active Ports.

- If you are using an **application level firewall** (for example Windows Defender), make sure that the following executables are enabled (from the MRTC application folder):
  - mserver.exe (the main service executable running the SIP, WebRTC and media stack)
  - MizuManage.exe (admin client)
  - lego.exe (used for Let's Encrypt auto TLS certificate)
  - MServiceHost.exe (supervisor)
  - \tsmodules\fs\FS_mserver.exe (PBX features and WebRTC processing in some circumstances)
  - tlsproxy_mserver.exe (TLS encryption)
  - mserverftp.exe (used to access recorded files)

MRTC can try to auto enable the above executables on the Windows firewall during install and first start, however if you are using some other firewall or if MRTC doesn't have enough permission to change your firewall settings, then you might need to add the above executables manually to your firewall exception list.

We recommend to turn off any external or third party port based firewall (you can still use the Windows built-in firewall where it is easy to set exceptions for applications and not for ports).

Gateway behind NAT

The WebRTC-SIP gateway can be used also behind NAT (located behind NAT or router, even without internet access).

Note: this section is not about WebRTC clients connecting from behind NAT. That is a normal use case and the gateway will handle those very well by default, without any special attention needed. This chapter is when you install the Gateway on a local LAN with SIP and WebRTC clients on the same NAT or connecting from the internet.

If your gateway is behind a NAT or Router, make sure to forward the ports above correctly if you need connectivity also from external network *(WebRTC clients or SIP server on the internet. If both of these are behind NAT, then there is no need to enable these ports except ports 80 and 443 if you need auto-generated SSL certificate).*
In this case make sure to set the "**Offer services for**" option correctly on the "Network" page of the Configuration Wizard.

- Both LAN and internet: select this if WebRTC clients might connect also from the public internet (make sure to set proper port forward on your router in this case)
- LAN: means all WebRTC clients will connect from local LAN, but the SIP traffic might be sent to the public internet (your SIP server is outside)

- Force LAN Only: means that all peers (including your SIP server and all clients) are located on the local LAN

If your gateway is behind NAT and no access to the internet or you don't forward ports 80 and 443 to it, then you will not be able to use its auto-ssl capability and no valid SSL certificate will be assigned. In this case you will not be able to use WebRTC from modern browsers, however old browsers or native WebRTC client apps might work just over unencrypted websocket. (use ws:\\gatewayIP). This is except if you use a webvoiphone.com subdomain since for this domain there is already prepared wildcard certificate for your convenience.

In other words, here are the **possible uses-cases**:

- Gateway on the public internet:
  You can just skip this chapter as you don't need to take care about NAT in this case. (WebRTC clients connecting from behind NAT are handled automatically)
  Set the "Offer services" option to "Internet only" if your SIP server is also located on the public internet or "Both LAN and Internet" of your SIP server or other SIP device is located on the same box or network.
  Your "Public IP" and "Bind IP" can be the same public address in this case.
- Gateway, SIP server and WebRTC clients all on local LAN:
  In this case you don't need to set any port forwarding in this case as all peers will be reachable directly. The only thing to pay attention to is the SSL certificate, since an SSL certificate can be only installed if you have a valid domain assigned to your gateway IP. To be able to use the auto-ssl functionality, you also need to forward 80 and 443 TCP ports except if you use a webvoiphone.com subdomain since for this domain there is already prepared wildcard certificate for your convenience.
  The Gateway can be used also without any certificate via unsecure websocket from browsers which doesn't require wss or from native WebRTC clients.
  Set the "Offer services" option to "Force LAN only" in this case.
  Your "Public IP" and "Bind IP" can be the same private address in this case.
- Gateway and WebRTC clients behind NAT, SIP server on the internet:
  If the SIP server or SIP service has good NAT handling capability, then everything should work just fine by default. Otherwise setup proper port forwarding on your NAT/router for the above mentioned ports.
  Set the "Offer services" option to "LAN" in this case.
- Gateway behind NAT, WebRTC clients on the public internet:

Set the “Offer services” option to “Both LAN and Internet” in this case and setup proper port forwarding on your NAT/router for the above mentioned ports.
Also set the “Public IP” to your network external IP.

If the gateway is behind NAT then the auto Let’s Encrypt certificate will work only if ports 80 and 443 are forwarded on your NAT/router. This is due to Let’s Encrypt policy, other ports can’t be used for this purpose.

Don’t run your WebRTC or SIP client from the same PC where MRTC is running. Most software has various bugs if uses the same local IP as the gateway IP!

Multiple SIP servers

You can use one MRTC gateway for multiple SIP servers (if you have more servers) for both outbound and inbound traffic.

Registrar routing:
By default all registrations will be routed to the default upper SIP server (what you have set during the configuration wizard or from global configuration search for “fwdregistrations_”).
If you have multiple upper SIP servers and the WebRTC clients need to register to different servers, make sure to set the routingforregister global config option to 1 (from the “Configuration” form).
Then the registrations will be routed after the routing rules which can be defined on the “Routing” form. The only exception is the called number since there is no called party in a registration (this is applicable only for calls).
However, if your SIP servers are for completely different domains/businesses, then you should consider to set-up a separate WebRTC-SIP gateway for each of them.

Note: The “routingforregister“ is only for registrations. Call and chat routing are always performed after your “Routing” settings

Call routing:
- Outbound calls are always routed based on routing rules which you define on the “Routing” form. Routing rules can be set for SIP servers (so you must create “SIP server” user(s) before to be able to work with routing.
- To be able to accept inbound calls (from your SIP server to WebRTC) you will need to create “Traffic Sender” user (usually with IP authentication with the Auth IP set to your SIP server address).
- Calls between users (such as WebRTC to WebRTC) are handled automatically.

You might also set the following global config options:

- routingforregister: 1 (to use Routing also for register requests)
- upperserverlookup: 2 (lookup upper server from tb_users sipservers. -1: guess (not for mizuonly servers), 0: no, 1: if no upper server was set in global config, 2: always)

More details about SIP servers can be found here.

Calls between endusers/extensions

Calls between endusers can be routed directly, without the need to route them via your upper SIP server(s), thus saving your server resources (CPU/memory).
These are usually calls between WebRTC clients (WebRTC to WebRTC), but they can be also WebRTC to SIP or SIP to WebRTC or SIP to SIP calls (if you register also SIP devices on the gateway).

This direct routing is enabled by default and it can be easily changed by the following global config values (from the "Configurations" form):
alloweuserusercalls: 0=no (don't check if local target),1=yes,2=disable (drop) call to endusers
fs_directroutrfstofs: -1=auto,0=no,1=via mizu server,2=within webrtc stack (route call between webrtc and webrtc within the webrtc stack), 3=quick websocket only,4=auto

To enable direct routing, set the alloweuserusercalls to 1 and fs_directroutrfstofs to 3.
To disable direct routing, set the alloweuserusercalls to 0 and fs_directroutrfstofs to 0.

If you disable direct routing, all calls will be routed via your SIP server(s). If the called endpoint is registered via the MRTC gateway, then the SIP server might route the call back to the gateway, so the call flow might look like this: Caller WebRTC endpoint -> MRTC gateway -> SIP server -> MRTC gateway -> Called WebRTC endpoint

Multiple registrations and call fork

One user can be registered from multiple locations at the same time and calls can be routed to all of its devices.
Use the following global configurations to modify this behavior:
fs_multireg: -1: default, 0: no, 1: yes (enabled/disable multiple contact address)
allowforkforsignaling: 0: no,1: partial,2: yes,3: yes and remember old addresses (send sip request to multiple recipients at once)

Connecting with SIP clients

You can also connect SIP clients via the WebRTC-SIP proxy, although these are usually connected directly to your SIP server since there is no need for protocol conversion
Register with two softphones and call from the first account to the second account.
Softphone configuration:

- domain: your gateway IP or domain name (and the SIP port if your changed it from the standard 5060 UDP port)
- proxy: you can leave it empty
- username / password: loaded from your SIP server (or from the Gateway if you created a special user with username/password authentication)

Check “Client configuration” from the “Help” menu for the exact details.

No other special settings are required (such as NAT, STUN, etc).
The network setting should be automatically handled by the server. If you don’t hear any voice you might change the RTP routing for the user(s) to “always route RTP” from MizuManage -> Users and devices -> Edit tab.

VoIP Push notifications

VoIP Push notifications are useful to notify sleeping or closed mobile and web SIP clients on incoming call (wake-up and handle the incoming call) and instant messages (message display).
Both the Apple PushKit and the Google FCM are supported.
Set the following global config options (MRTCAdmin -> Configurations form) to enable push notifications in your server or gateway:

- pushnotification: 2
- pushnotification_chat: 2

If you are using the Mizu webphone or softphones, push notifications are supported by default. If you are using a third party SIP client or develop your own, then you need to add client support for it in your app to make it working. The process is described here.

Webserver configuration

In case if you are using a web based WebRTC client, you will have to host it on your Web server such as IIS or Apache.
However the WebRTC functionality has nothing to do with your Web server. Your Web server will be used only to serve your web content (including the html/css/JS needed for your WebRTC client),

however once launched in a browser, WebRTC clients will connect directly to the WebRTC gateway (first with websocket, then also via ICE/DTLS/SRTP while in call).
Thus the WebRTC and **the WebRTC gateway have nothing to do with your Web server**.

STUN

The WebRTC-SIP proxy has a built-in stun server **enabled by default**, usually listing on port 8080.
The stun server is listening on a port calculated at startup in the following way:
-mainaportudp if that is set
-mainaport if that is set and if not 80
-otherwise on port 8090
You can easily get the STUN configuration required for your WebRTC clients from Help menu -> "Client configuration".
The built-in stun service can be disabled by setting the quickstun to 0 and the stunserver_run to 0, however this is not recommended.
You can also configure your WebRTC clients to use another STUN service, however this doesn't make much sense as the built-in STUN in the MRTC Gateway is just perfect for WebRTC.

TURN

The Mizu WebRTC gateway has a built-in TURN service **enabled by default**.
The TURN service is listening on the same port which you have set for websocket and API.
You can configure this port from the Configuration Wizard -> Network page -> Access port
or by just changing the "mainaport" value in the global configuration ("Configuration" form).
To change any turn related configuration, search for "turn" on the "Configuration" form. However, these should be rarely required because the default settings are just fine for all networks.
The Gateway also handles the TURN authentication automatically, allowing only the WebRTC clients which are successfully connected via your Gateway.
You can easily get the TURN related configuration required for your WebRTC clients from Help menu -> "Client configuration".
You can also configure your WebRTC clients to use another TURN service, however this doesn't make much sense as the built-in TURN in the MRTC Gateway is just perfect for WebRTC.

Codec

The gateway has support for all the commonly used audio and video codecs:

G.711, OPUS, G.729, G.723, GSM, iLBC, Speex, G.722, VP8, H.264
The codec is negotiated automatically for each call depending on WebRTC client and your SIP server capabilities and other circumstances such as the available bandwidth.
The codec used for calls can be also influenced by the “choosecodecs” user configuration (which can be also set on the “Users and devices” form -> Functions page -> Allowed codec setting).
The codec to be available for the WebRTC endpoints can be set with the “fs_codec” option whose default value is PCMU,PCMA,OPUS,GSM,G722,VP8,H264. Usually there is no need to change this since a browser WebRTC client has support only for PCMU, PCMA and OPUS audio and VP8 or H264 video.

Transcoding

By default, the gateway will try to avoid transcoding when possible by negotiating a common codec between caller and called parties. If necessary, it will convert automatically from typical WebRTC codecs (OPUS and G.711) to typical telecom codecs (G.729, GSM and others). Change the following only if really needed.
If one of your peers has limited codec capabilities or the accepted codec(s) doesn’t match with the sender codecs, then set its “needcodecconversion” to 1 (for the target user which is usually a SIP Server user or Enduser). This can be also controlled on the "Functions" page of the "Users and devices" form.
Change the “convertcodecs” global config value to the target codec payload list. The default value is 0,8,18 which means PCMU,PCMA and G.729.
To force always codec transcoding you can set the “fs_transcode” option to 2 (And set to 0 to avoid transcoding whenever possible).
The server is able to transcode between the following codecs: G.711 A-law , G.711 A-law ,G.729, G.723.1, GSM, OPUS, Speex 2,3,4,5,6 (narrowband, wideband and ultrawideband), G.726 and G.722.
You might also have to set the “choosecodecs” field for the target user (same as the “convertcodecs” global config value) and the “convertcodecsforced” global config option to true.
Be aware that codec transcoding requires a high amount of CPU usage. For example one CPU (core) can handle around maximum 30-70 simultaneous transcodings between PCMU and G729 on full load.

How to avoid transcoding?

Codec transcoding should be avoided whenever possible because it will increase the CPU usage and also will degrade the quality a bit. By default the MRTC gateway will try to negotiate a common codec between the endpoints and transcode only when strictly necessary.
The good news is that all common WebRTC clients (including browser WebRTC stacks) and SIP servers have G.711 (PCMU and PCMA) capabilities.

To avoid codec transcoding, make sure to enable PCMU and/or PCMA on your SIP server.
If your SIP server has support for OPUS, then enable this also as its wideband variant has much better quality then G.711.

Optimal codec settings

The MRTC WebRTC-SIP gateway is fine-tuned by default for best call quality.
This means that when possible, it will use a wideband codec (usually OPUS as this is commonly supported by WebRTC and it has a very good quality, much better than PSTN).
However, OPUS doesn't make sense if the call has to be routed to PSTN, because PSTN usually supports only narrowband codecs.
Usually WebRTC to WebRTC calls are done with OPUS.
If your SIP server has support for OPUS, then calls to SIP will also use this codec.
If a common wideband codec can't be found, then it will use a narrowband codec (usually G.711 since this is supported by both SIP and WebRTC without the need for any transcoding).
If no common codec is found then it will transcode.

To achieve the best possible call quality, do the following:

1. Enable OPUS on your SIP server if it has support for this codec, especially if there are calls between users/extensions such as WebRTC to SIP or SIP to WebRTC calls. If all calls will go to landline or mobile (PSTN or a VoIP carrier, then OPUS is unnecessary, but it doesn't hurt if it is enabled for occasional calls between users within your network)
2. It is highly recommended to always enable G.711 (PCMU and/or PCMA). Actually this is enabled by default in every software, so just make sure to don't disable it. Make sure that your carrier (if any) also has support for G.711, so the calls can pass end to end without the need for transcoding (most carriers allows G.711)
3. If, for some reason you must send out a call (or receive calls) with other codecs (such as G.729) the MRTC Gateway will do the transcoding automatically, but this will have a cost in the performance and might degrade the call quality a bit. You can also force/disable or change the transcoding settings after your needs.

Slow call setup

With some WebRTC clients you might experience a considerable delay at call connect (call connecting/setup/in progress). This delay is usually caused by poorly written or configured WebRTC clients and it has nothing to do with the MRTC gateway. To solve the problem, check your WebRTC

client configuration and set a timeout for ICE/TURN/TURN (or lover the existing timeout value). Contact the developers if it doesn't have such configuration.
Around 4 second timeout should be enough in all circumstances (media candidates not collected within 4 seconds are likely to be invalid anyway).

Call recording

The MRTC gateway has full call recording support which can be enabled/disabled per user or system wide (both can be done by just toggling a checkbox on the admin user interface). Then you can easily manipulate the recorded files (search, convert to mp3, export, access in the file system, access via API, etc).

You can set the system-wide call recording from Config menu -> Configurations -> Recording -> Call.

The voice recording option can be set for individual users by checking the "Voice Record" checkbox on the user configuration form in the MRTCAdmin (Users and Devices -> Functions tab). The followings fields will be affected in the tb_users table:

- record: 0 means no, 1 means yes
- voicerecupload: optionally you can specify a HTTP or FTP URL where the recorded files will be uploaded

Conversations will be saved in the directory specified by the "serverftpvoice" global config option.
The exact location will be: serverftpvoice\databasename\currentday\voice.xxx
A separate backup can be created in the directory specified by the "voicebackupdir" global config option.

Old files can be deleted by setting the "keeprecorded" option accordingly (days to keep).
Recorded files are compressed and encrypted by default.
From the MRTC Admin you can work with the recorded calls from the "CDR" form.
Recorded conversations can be played on the "CDR" form (Select the "Recorded Conversations" radio item, select the desired record and click on the Play button) or from the "Voice Record" form.
You can also export the files as wav or mp3.
Users can replay the last record by the following DTMF digits: *4*

You can also use the voicerecdownload API to download any recorded file(s).
The request should look like this:

https://yourdomain.com/mvapireq/?apientry=voicerecdownload&callid=XYZ&maxrecords=1&authkey=SERVERKEY&authid=ADMINUSERNAME&authpwd=ADMINPASSWORD

In case if your are using the Mizu webphone, then the HTTP or FTP upload URI can be also configured in the webphone itself using the webphone voicerecupload parameter or voicerecord API.

Call recording will use extra disk space, can make the routing and media path longer by disabling peer to peer routing and also will increase your server I/O and CPU usage.

Technical details below:
System-wide call recording can be set from Config menu -> Configurations -> Recording -> Call or you can use the following SQL updates:

To disable recording for all users, execute the following SQL's in the "Direct Query" form:
update [tb_users] set record = 0, voicerecupload='', RouteRTPCaller = 1, RouteRTPCalled = 1 where type in (0,5,9)
update tb_settings set valstr = '1' where keystr = 'defroutertp'
ALTER TABLE [tb_users] DROP CONSTRAINT [DF_tb_users_record]
ALTER TABLE [tb_users] ADD CONSTRAINT [DF_tb_users_record] DEFAULT 0 FOR [record]

If you wish to enable call recording for all users, execute the following SQL's in the "Direct Query" form:
update [tb_users] set record = 1, RouteRTPCaller = 7, RouteRTPCalled = 7 where type in (0,5,9)
update tb_settings set valstr = '7 ' where keystr = 'defroutertp'
ALTER TABLE [tb_users] DROP CONSTRAINT [DF_tb_users_record]
ALTER TABLE [tb_users] ADD CONSTRAINT [DF_tb_users_record] DEFAULT 1 FOR [record]

To force recording for all calls (even those where the media would be routed peer-to-peer otherwise), set the following global config options (This is usually not needed because even without these settings, calls between the WebRTC clients and your SIP servers will be recorded so these settings affects only WebRTC to WebRTC calls):
defroutertp=7 (force RTP routing by default)
alloweuserusercalls=0 (optional setting to disable direct calls and route all calls via the upper SIP server –usually not needed)
fs_directroutrfstofs= 1 (disable fast path routing)
fs_forceoffrouteforvoicrec=4 (turn off webrtc direct routing. 0=no,1=yes on low load,2=yes-auto,3=yes, 4=force for all calls)
hasturn=0 (this will disable to TURN server as this doesn't support call recording. TCP candidate will be still available if needed)
disablep2prtprouting=1 (disable peer to peer media path detection)
Delete the following routing rules if exists: check_user, check_user_notregistered, check_user_registered.

Note: MRTC versions released after 2017 November automatically detects when recording is needed between two WebRTC endpoints and disable fast direct routing automatically.

Chat recording

The users IM history can be recorded and stored in server database, tb_messages table.
For this set the "logmessenger" global config option: 0=no,1=on low load,2=always

You might also disable fast message forwarding by setting the "fastmessagequeue" configuration option to 0.
The recorded messages can be seen on the "Chat Logs" form MRTCAdmin.

Note: In MRTC versions released after 2018 Marc, you can easily control the chat recording from Config menu -> Configurations -> Recording -> Chat.

DTMF

The WebRTC-SIP Gateway will automatically handle DTMF sent from your client and server and it is capable to also transcode them if needed.

- By default will accept the DTMF digits as RFC 2833 or as SIP INFO methods from both your WebRTC clients and SIP servers.
- By default it will send the DTMF digits as SIP INFO methods or as received but it might be capable to failover to RFC2833 or InBand if the remote endpoint doesn't support or understand SIP INFO

You just need to change the fs_dtmf_type_extern global config option to change the dtmf mode.
For this, open the MRTCAdmin -> Open the "Configurations" form (below the "Other" node in the tree view) -> enter "fs_dtmf_type_extern" in the search box -> Change the value after your needs -> Restart the gateway.
You can restart the gateway from MRTCAdmin Control menu or from windows services restart the mserver service.

Your SIP server -> MRTC Gateway -> WebRTC client:
Both RFC2833 and SIP INFO are accepted from your gateway, so there is no reason to change this.

WebRTC client -> MRTC Gateway -> Your SIP server:
To set the DTMF method to use from our gateway to your SIP server, use the following settings:

- For SIP INFO: set the fs_dtmf_type_extern to 1 (default)
- For RFC2833: set the fs_dtmf_type_extern to 2

In-Band dtmf is not recommended.

Other important dtmf related global config options:

- handledtmfevent: enable dtmf based features such as transfer or call. 0: no, 1: auto (default), 2: always, 3: even while ringing (set to 2 if to always force handle dtmf events)

- forwarddtmf: 0: no, 1: auto (default), 2: yes
- convertdtmf: -2=never convert, -1: auto (default), 0=no, convert,1=inband,2=info, 3=rfc2833,4=inband+orig,5=info+orig,6=rfc+orig,9=suppress
  This can be set also by user (tb_users.confertdtmf field)
- You can find all the DTMF related settings if you search for "dtmf" on the "Configurations" form.

Logs

You can find the Gateway logs from Files menu -> Folders -> Server log directory. Use a fast text reader to work with the files such as F3 from Total Commander.
The logs files are managed automatically by the gateway (auto deleting old log files or if you are low on disk-space).
You can change the server log level from Control menu -> Log level.

Integration

You can integrate the service with your backend or web service using direct database access or API.
The below documentations are for the softswitch, but most can be also applied for MRTC (except the billing related sections).

- Integration guide
- Database access
- API

For example you can use the addserver API to add an upper SIP server:
Possible parameters:

- u_type: 0: both, 5: only as traffic sender, 9: only as SIP server (use 0 to add upper servers)
- u_name: name (either the name or the username must be provided)
- u_username: username (either the name or the username must be provided)
- u_password: will be auto generated if empty
- u_domain: domain name (either the domain or the ip must be provided)
- u_ip: IP address. it can be also IP:port (either the domain or the ip must be provided)
- u_port: port number if not set as part of the domain or IP (default is 5060 if not provided)
- u_proxy: outbound proxy (optional)
- u_transport: udp, tcp or tls (default is udp if not set)
- u_auth: the needauth db field (default is 1 which means IP based auth; check the documentation for other possibilities)
- u_email: optional

- u_country: optional
- u_currency: optional
- u_credit: optional
- u_routing: routing entry id or name (set to "null" if no routing entry is required; set to empty to auto-guess).
- u_routingpriority: routing priority
- fieldnameX / fieldvalX: optional extra fields to be set in tb_users (X can go from 0 to 99)

Here is a very simple example:
http://10.2.0.71/mvapireq/?apientry=addserver&authkey=11979586&authid=voip_admin&authmd5=ccb612764b35204ed0d212bb80ef931a&authsalt=7394106&u_name=SERVERNAME&u_domain=DOMAIN&now=711

The same can be done also with SQL queries.
In this case you just need to insert two records to tb_users (traffic sender and SIP server) and add the SIP sever to tb_routinglist.

Add traffic sender:

```
insert into tb_users (username, name, password, needauth, type, candial, belongstocompany, domainname,
authip,port,protocol,RouteRtpCaller,RouteRtpCalled,MaxLines,maxlinesoffpeak,comment) \
  values ("'NAME', 'NAME', 'RANDOMPASSWORD',1, 5, 1, 'DOMAIN', 'IP', 5060,0,1,1,1000,1000,'ANYCOMMENT')
```

- the type field must be set to 5 for traffic senders
- needauth must be 1 for source address based authentication
- protocol is 0 for UDP, 1 for TCP, 2 for TLS

Add SIP server:

```
insert into tb_users (username, name, password, needauth, type, domainname, ip, proxyaddress,
port,protocol,RouteRtpCaller,RouteRtpCalled,MaxLines,maxlinesoffpeak,comment) \
  values ('NAME','NAME','RANDOMPASSWORD',1, 9, 'DOMAIN','IP', 'PROXY',5060,0, 1,1,1000,1000,'COMMENT')
```

- the type field must be set to 9 for SIP servers

Security

When using the SIP stack in “gateway mode” then it will not store your VoIP user credentials (SIP passwords). All other sensitive tasks such as billing and user management is also done entirely by your server (no such data is stored on the gateway) thus MRTC is not a critical component from the security point of view. However the list of the users (the SIP usernames) are always stored also on the gateway thus you can’t leave the gateway entirely unprotected.
Windows OS is as secure as any other (Linux/MAC/BSD) if you are using a clean install only to host the WebRTC gateway service with the built-in firewall turned on (allowing only the Gateway process). This

is because all Windows services are turned off on such a server and kernel vulnerabilities exploitable via basic TCP/IP are extremely rare (not more than in Linux or even BSD).
From security perspective it is important to run the MRTC gateway on a clean OS with no any unsafe third-party software installed.

The WebRTC gateway is based on our Softswitch core which implements a long list of security features applicable also for the Gateway. This includes transport and session based limits, rule based filtering, rate limiter, auto-ban, auto-blacklist, call limits, fraud detection, DDoS protection and more.
For the details please read here.

High availability

HA, failover and redundancy is supported for both downstream and upstream.
You can use a second backup gateway, multiple gateways for the same SIP server, handle multiple SIP servers by the same gateway or for the highest availability: use multiple gateways with multiple (same or different) SIP servers.

Client side failover:

There are multiple possibilities to implement HA toward the WebRTC clients:

- **API**
  The best option is to implement a simple availability verification into the client software and failover to other instances if no answer or bad answer have been received. This can be done by using the wsload API which will return one of the following answers:
  - ERROR, errortext indicates gateway malfunctioning
  - OK: WS Load: X type … indicates that the gateway is available. The X is a number from 0 to 4 indicating the gateway load (0-low, 1-normal, 2-high, 3-very high, 4-extreme). You can treat 4 as gateway malfunction, otherwise you should prioritize the gateway returning the lowest number.
    If your gateways are geographically distributed, then you can check the response time to find out the nearest one (or more correctly, a combination of the gateway load and response time should be used).
- **UAC failover**
  Some WebRTC-SIP client might already have a failover mechanism implemented on the transport layer.

A simple way to handle gateway availability is on the SIP protocol level. Failover if there are no or bad response of the REGISTER (or OPTIONS/INVITE requests)

- **Network**
  Another way to implement failover is to simply redirect your WebRTC-SIP clients to other gateways on failure. This is not so sophisticated like the above mentioned methods, but offers administrators a simple way to circumvent major outages. Failover can be forced by the following methods
  - domain name redirect (easy usage but propagation will take some time. You need to set also the A record as SRV records usually can't be used from browser clients)
  - BGP (requires some expertize and access to BGP, thus mostly available only for ISP's)
  - dynamic IP (most server hosting providers nowadays can provide dynamic IP service which can be used instead of domain redirection as an instant redirect for all peers)

Note: the mizu webphone implements both API based and transport layer failover if you configure it with at least two gateways

<u>Server side failover:</u>

This is about auto-prioritization among upper servers (your SIP servers).
You can configure multiple upper servers for a gateway and define various routing rules such as simple load balancing, quality based routing named (BRS) or any specific routing rules (depending on caller, called, time and other criteria). Check the routing guide for a quick description.
Beside the routing rules, the MRTC gateway is capable to auto-detect malfunctioning servers and automatically deprioritize them. This is part of the BRS algorithm, but you have a basic failover even if you are not using BRS (basic failover meant to detect bad routes and deprioritize them if there are better working SIP servers).

MRTC implements failover on all levels:

- Basic failover ("hard" failover, mentioned above, based on previous upper server behavior)
- Routing automatic ("soft" prioritization, discussed above, based on detailed statistics)
- Routing rules (upper server prioritizations/failover after pattern and/or destination priority)
- Re-Register (redirect the registration to a working server on bad or no response)
- Re-Dial (in-call failover when a bad or no response is received from the upper SIP server)
- Or simply you might use a separate MRTC gateway for each of your SIP servers

More details can be found here, here, here and here.

Feature support

Certain features (such as voice recording) doesn't depend on any external support, however some features require support also by the WebRTC client and/or your SIP server (and/or SIP peers):

- For IM/chat to work, your WebRTC client and SIP server need to support the SIP MESSAGE standard which is the basic instant messaging protocol as described in RFC 3428. Note that messaging can work between the clients on the same gateway even if your server doesn't support RFC 3428.
- For DTMF to work your SIP server and WebRTC client must have support for RFC 2833 or SIP INFO. The gateway can also perform the conversion between these if needed. In-band DTMF is routed as-is, so this method will also work if both your SIP server and WebRTC client use this method. The best is to make sure that your server can send/receive the DTMF in the same format that is emitted/accepted by your WebRTC clients.
- For presence to work your software needs support for PUBLISH (required/used only by WebRTC) and for SUBSCRIBE/NOTIFY as specified in RFC 3856.
- For video to work your SIP server and/or SIP peers need to support video codecs such as H.264, H.265, VP8, VP9.
- For call transfer you will need support for SIP REFER as described in RFC 3515.
- For call forward you will need support for SIP response code 301 (Moved Permanently) and/or 302 (Moved Temporarily).

In general, WebRTC clients tends to has less features then SIP clients, but the most important call features should be compatible regardless of the transport layer.

If some feature is supported by your WebRTC client but not by your SIP server, then it will work only for WebRTC->WebRTC calls (and not for WebRTC->SIP or SIP->WebRTC).
The MRTC gateway also does its best to compensate on missing support. For example, the chat might work even without SIP MESSAGE support by carrying the message by SIP OPTIONS or SIP NOTIFY requests if your SIP server allows. Also the basic presence might work without SUBSCRIBE/NOTIFY support by using the peer registrar state.

Extra features

Most of the extra SIP features will be just forwarded transparently by the gateway and actually negotiated between the WebRTC clients and your SIP server. Most SIP features will work out of the box by default. You might read below only if some extra or call control feature doesn't work for you either because of no direct mapping between SIP and WebRTC or lack of implementation in your WebRTC client or in your Softswitch.

Class 5 features should be handled on your IP-PBX or Softswitch, however the Gateway also has some built-in features which might be useful for you (for example voicemail or call forward) assisted or handled entirely by the WebRTC gateway. Also for some functionalities there are no direct mapping between the WebRTC and SIP protocol so they can be solved only by extra features provided from the MRTC Gateway, by the Softswitch or by the WebRTC client (for example conference rooms). Other features don't depend on the protocol used (no direct support in SIP or WebRTC) but can be implemented on the server side using some separate protocol (usually via a HTTP API exposed to clients).
Some extra PBX features can be enabled by setting the "fs_pbx" global configuration option to "1" (or disabled by setting the same to "0").
Here are a few of the extra features:
RTMP: as a bonus, MRTC includes also a flash gateway module so it is compatible also with flash based voip clients. The RTMP is listening on port indicated by the "hasflash" setting defaulting to 1939. Disable by setting to 0.
Call forward: can be enabled from users and devices form -> functions tab
Call transfer: available as specified SIP standards (via all devices with transfer support) or via *9*number# dtmf
Voicemail: Enable/disable from the "voicemail"global config. Then enable/disable per user from "Users and devices" form -> "Functions" tab. Default access number is 5000 (with pin) and 5001 (with auto authentication). Auto-email forward is enabled by default.
Conference: via SIP standard, via dtmf *1*number# or via conference rooms using extensions between 5100-5199 for narrowband and 5200-5299 for wideband. You can use the "getconfroom" API to auto allocate a conference room and its access number for the user.
Special numbers and IVR's such as music, record/playback, vide record/playback and others
Missed call notifications by email
Barge in: via the "Voice here" form, or right click on current call or 5009 access number
Softphones: Mizutech can also offer customized/branded softphones which works with the Gateway or directly with your SIP server: Browser webphone, Windows softphone, Android softphone, iOS softphone, Symbian dialer, Other softphones
IVR: The IVR module can be used for various tasks like access numbers, callback, customer support etc. You can assign different IVR's to different access numbers by using the "**Campaigns**" form. To create a new campaign, just click on the + sign and enter a "name" for the new record. The most important configuration for an IVR campaign is the script. Switch to the "details" tab to select a "Script". Scripts can be created by using the "IVR" form. The gateway is shipped with several preconfigured script examples, but you should easily add new scripts or modify the existing ones by following the admin guide or the IVR documentation.
SMS: by default the gateway is capable to route SMS message as SIP MESSAGE (RFC 3428). However, it is also possible to use an external service which can be accessed by a HTTP API. Just set the "smsurl"

smsurl global config option or create SMS GW endpoints if you wish to use more than one provider. A guide can be found here.
API: the Gateway also has a HTTP API which can be used for various tasks. See the API documentation for the details.
Many other features are enabled for you by default and some of them can be changed/fine-tuned from configurations, such as caller-id, ring groups, call hold, call waiting/park/pickup and others.
The WebRTC-SIP proxy is based on the VoIP Server core thus you can also access almost all of the Softswitch features. See the Softswitch documentation for the details.

If you are not sure where to find a specific configuration option, search for your keyword in:

1. "Configurations" form within MRTCAdmin
2. This guide
3. VoIP server guide (most of the settings can be applied also for the MRTC gateway)
4. Check the other documentations (some of them are relevant also for MRTC)
5. Or ask Mizutech support

FAQ

How to get my WebRTC-SIP gateway

Follow these steps:

1. Have a look at the home page and eventually download and try the free WebRTC-SIP gateway.
2. If seems to be ok for your needs, purchase from here.
3. On your payment Mizutech will send the download link by email (you can just reinstall to upgrade any existing versions without data/settings lost)
4. Contact mizutech support for any help or if you run into any issue using the MRTC gateway.

Can't connect/register

1. Make sure that the service is started (NT Services "mserver")
2. Make sure that the required ports/executables are not firewalled
3. If the server is running on a private IP (local LAN) and you wish to connect from external network, make sure that the required ports are forwarded on your router
4. Check the Dashboard, Analyze and Logs forms for any errors

5. Make sure that your IP/domain configuration is correct (bindip should be present on the local interface, localip should be your box public IP, the configured domain DNS A record should be set to the localip)
6. Make sure that TLS is configured correctly. If you enabled auto TLS from the configuration wizard, then you can see related issues in the logs if any.
   Test TLS connectivity from MRTCAdmin -> Config menu -> Utilities -> Tests -> TLS Certificate test -> TLS Proxy Test
7. Don't use special UTF characters in username/password (some SIP servers don't like non-ASCII characters).
8. Make sure that you are using a correct SIP username/password valid on your SIP server.
9. Some server require to set a different username/auth username (Username is usually the Extension ID). If you are using the Mizu WebPhone, then set the Extension ID as the webphone "Caller ID" ("username" parameter) and set the Auth user name as the webphone "Username" ("sipusername" parameter).
10. Check if otherwise you can connect with a simple SIP client (first directly to your server, second to the gateway). If this works, then the problem is WebRTC or WebSocket related.
11. Test if the gateway WebRTC service is listening from MRTCAdmin -> Config menu -> Utilities -> Tests -> TLS Certificate test -> WSS SIP Test
12. Check the browser console logs and the gateway logs (MRTCAdmin -> Files menu -> Folders -> Server logs).

Can't make calls

1. Check the Dashboard, Analyze and Logs form for any issues
2. Make a test call first from a simple SIP client such as mizu softphone or x-lite (first directly to your server, second to the gateway). If this works, then the problem is WebRTC or WebSocket related.
3. If you are making call to a local user, make sure that the called user is registered when you call it
4. Check the CDR disconnect reason (if the call reached the gateway) from MRTCAdmin -> CDR form.
5. Make sure to allow Media Permissions for WebRTC in your browser.
6. Check the browser console output.
7. Check if the call reaches the gateway and inspect the gateway logs (MRTCAdmin -> Files menu -> Folders -> Server logs).
   a) Open the last logfile ("log_xxx.dat") with a fast text viewer (For example F3 from TotalCommander). To find application errors, open the last log file in the server app directory and search for "ERROR" and "WARNING". To find a call, search for "INVITE sip:callednumber".

   b) In case if the logs doesn't contain the SIP signaling details, enable the detailed logs first (Control menu -> Logs -> Set -> On), reproduce the problem and check the log files again

8. Check if the call reaches your SIP server and inspect your server logs.

Incoming calls not received

1. Check if the call reaches your SIP server. If not, then the problem has nothing to do with WebRTC.
2. Check if the call reaches the WebRTC gateway. In not, then make sure that the WebRTC client is registered to your server at the time when the inbound call is routed to it.
3. Check if the call reaches the WebRTC client (check the browser console output with detailed trace/log level, search for "INVITE sip").

Calls are disconnecting

- Call disconnection immediately upon setup can have many reasons such as codec incompatibility issues, NAT issues, DTLS/SRTP setup problems or audio problems.
- Call disconnection after 20 seconds might be caused by MEDIATIMEOUT and MEDIATIMEOUTSTART global config option if there is no audio (RTP stream) or one way audio only.

No voice or one side audio

Change the RTP media routing for one or all users.
For one user this can be changed from the "Users and Devices" form -> Functions page -> Media relay.

For all users it can be changed in the following ways:

To disable media routing, execute the following SQL's in the "Direct Query" form:

```
update [tb_users] set RouteRTPCaller = 1, RouteRTPCalled = 1 where type in (0,5,9)
Update tb_settings set valstr = '1' where keystr = 'defroutertp '
ALTER TABLE [tb_users] DROP CONSTRAINT [DF_tb_users_record]
ALTER TABLE [tb_users] ADD  CONSTRAINT [DF_tb_users_record]  DEFAULT 0 FOR [record]
```

To enable media routing, execute the following SQL's in the "Direct Query" form:

```
update [tb_users] set RouteRTPCaller = 7, RouteRTPCalled = 7 where type in (0,5,9)
update tb_settings set valstr = '7 ' where keystr = 'defroutertp'
```

```
ALTER TABLE [tb_users] DROP CONSTRAINT [DF_tb_users_record]
ALTER TABLE [tb_users] ADD  CONSTRAINT [DF_tb_users_record]  DEFAULT 1 FOR [record]
```

To force the media routing even for WebRTC-WebRTC calls, also set the following global config options:

```
defroutertp=7 (force RTP routing by default)
alloweuserusercalls=0 (optional setting to disable direct calls and route all calls via the SIP server –usually not needed)
fs_directroutrfstofs=1 (disable fast path routing)
fs_forceoffrouteforvoicrec=4 (turn off webrtc direct routing. 0=no,1=yes on low load,2=yes-auto,3=yes,4=force for all calls)
```

Long call setup time

If you experience long setup or ring time, that is almost always due to poor client side WebRTC implementation. Make sure to use a WebRTC client with proper ICE timeout limitations such as the mizu webphone. It is normal for ICE queries to take 20 or more seconds, however it is not normal to wait for these to complete on the client side as such delayed responses are always useless and mostly caused by unused network interfaces.

Chat doesn't work

Make sure that your SIP server and WebRTC client has support for SIP MESSAGE as described in RFC3428.
Most old Asterisk installations might not have support for this by default. You might use Kamailio for this purpose or any other softswitch (most of them has support for RFC 3428).
For new versions of Asterisk based SIP servers you need to properly configure SIP MESSAGE routing in order for IM to work.
See also Asterisk patch or FreePBX settings.

Presence doesn't work

Presence is implemented as standard SIP PUBLISH/SUBSCRIBE/NOTIFY so your SIP server must support RFC 3265 and RFC 3856 in order for this to work.
For Asterisk based servers you can find configuration details here and here.

Video doesn't work

Make sure that your server has full support for H.264 and VP8 or it bypass the media routing and forwards the SDP attributes correctly.

DTMF doesn’t work

WebRTC clients are usually sending the DTMF with SIP INFO so make sure that your SIP server can handle this DTMF method.
Adjust the gateway global configuration after your needs:

- fs_dtmf_type_intern: //dtmf-type to clients : 0=default, 1=info, 2=rfc2833
- fs_dtmf_type_extern: dtmf-type to servers: 0=default, 1=info, 2=rfc2833
- fs_dtmf_duration: dtmf-duration. def: 600
- for other DTMF related settings, search for “dtmf” in the global configuration

For more details, read here.

DTMF triggered actions

The gateway has built-in functions to be triggered by DTMF digits, useful for endpoints with no conference or transfer support.
For these to work the PBX module have to be enabled and the handledtmfevent global config option must be set to 2.
Here are the defined keys:

- create conference or add new user to conference: *1*number#
- unattended transfer: *9*number#
- transfer with consultation : *8*number#
- talk with new client while in transfer: 1
- talk with old client while in transfer: 3
- disconnecting last conference party: *5*
- disconnecting conference party: *6*number#

How to explicitly set the IP address used in signaling

The RTC gateway might auto-detect another (best match) address to be used in the SIP signaling even if you have set a bindip and/or localip. This might be the case on servers with multiple network interface or a different/changing external IP.
You can use the following global config options to prevent this:

- autouselocalip: set to 0
- autodetectlocalip: set to 0

Additional related parameters:

- localip (you might set this explicitely)
- bindip (you might set this explicitely)
- localinternalip (you might set this explicitely)
- nathandling
- nataddrtype
- usephisicalfromaddr

You can also explicitly set the ip address to use per peer by setting the interface field in tb_users and the secondarybindip values in the global configuration.

I don't see some calls on my SIP server

The WebRTC gateway by default will route the calls between the endusers bypassing your SIP server with the advantage of freeing up resource utilizations on your SIP server.
You can disable this behavior by setting the alloweuserusercalls and fs_directroutrfstofs global config options to 0 thus forcing all calls to be routed via your server.

How to forward custom SIP headers

The gateway might drop or rewrite SIP headers in the SIP signaling, however you can add any custom SIP headers with the "X-" prefix to be forwarded as-is.
Example:

```
X-MyHeader: 1234
```

How to restrict users max concurrent calls

This can be configured globally from the Configuration Wizard "Users" page or per user from "Users and Devices" -> "Functions" page Max Lines settings.

In case if you need to set later for all users, you can use the following query (copy-paste to Direct Query form, replace X after your needs and execute):
For individual users:

```
update tb_settings set valstr = 'true' where keystr = 'checkmaxlines'
update tb_settings set valstr = 'X' where keystr in ('endusersdefmaxline', 'maxlineforautotunnelusers')
update tb_users set MaxLines = X, MaxLinesOffpeak = X where type = 0
```

For trunks (all traffic from/to your SIP server):

```
update tb_settings set valstr = 'X' where keystr = 'maxlinefornewunknownparents'
update tb_users set MaxLines = X, MaxLinesOffpeak = X where type > 0
```

Note: the total number of maximum simultaneous calls might be also restricted by your MRTC license.

Call quality problems

Call quality is influenced primarily by the followings:

- Network conditions (QoS)
- Codec used to carry the media (wideband has better quality)
- AEC, AGC and denoise (enable/disable these in your client after your needs)
- Hardware: enough CPU power and quality headset/microphone/speaker (try a headset, try on another device)
- OS/Browser (drivers problems)
- Settings (codec, jitter size, AEC, etc)
- Software problem or bug
- VoIP route (carrier/call termination/voip service provider call quality)

If you have call quality issues then the followings should be verified:

- Check your network quality: speed, packet loss, delay and jitter here. (Other tools: a , b, c, d)
- Check whether you have good call quality using a third party softphone from the same location (try X-Lite for example). If not, than the problem is not related to the gateway (continue to look for server, termination gateway or bandwidth issues)
- Make sure that the CPU load is not near 100% when you are doing the tests
- Make sure that you have enough bandwidth/QoS for the codec that you are using (G.729 requires around 24 kbits. G.711 requires around 80 kbits. Opus requires around 20-50 kbits. However the overall bandwidth is rarely the problem and you should check if you have these minimums constantly, with less packet loss then 2% and no big variations in roundtrip delays)
- Change the codec

- Try to set only one single codec (such as PCMU) to make sure that the problem is not caused by poor server side handling of the automatically selected codec
- Perform a network trace (With Wireshark for example; Filter for SIP and RTP. Check missing or duplicated packets)

Optimize for best call quality:

- If possible, avoid RTP routing on the server side altogether. This will allow peer to peer direct media. If your SIP trunk provider requires the RTP packets to be sent from your IP, then configure your server with RTP routing only for outbound/inbound calls (and not for user to user calls)
- Use (enable) OPUS (and speex) wideband if possible. Wideband codec has much better quality then narrowband and even if your service providers(s) accepts only codec like G.729 or G.711, this can be auto-negotiated and you should not force only these codec's on your server (allowing wideband for user to user calls)
- If possible, avoid codec conversion (transcoding). End to end G.711 has better quality then half-way OPUS and then conversion to G.711
- Follow best practices for VoIP (make sure that you have enough and good quality bandwidth with minimal jitter and no packet loss, make sure that your server resources are not over utilized and the CPU is cable to handle RTP routing with minimal delay, etc)

Backup/restore

Just copy the mserver.sdf file for a full backup.

Details:
Automatic backup can be configured from the configuration wizard.
In case if you wish to perform the backup with other tool or manually, then all you need to copy is the mserver.sdf file. Stop the service and close the MRTCAdmin client first to release the database.
This is the internal database, which contains all the settings and data such as routes and CDR records.
All the other files can be recreated by simply reinstalling the gateway with it's standard installer. (Once reinstalled, just replace the .sdf file in the app folder with your backup).
You might also backup the other files if you wish, in this case you might exclude the logs.
You might also backup the call recording files separately, which are stored in the "recorded" subfolder by default.

Change upper SIP server address

This is about changing your SIP server IP address or domain name (the upper SIP server where the calls are routed and received from) if you are using a single SIP server.
For example if you migrated your SIP server to a new location or if you have to change the upper SIP server details for any reason.
In case if you are using a licensed MRTC gateway, make sure that your license allows the usage also with the new SIP server address.

GUI configuration (this will not rewrite the upper server for existing users):
You might go trough the Configuration Wizard again and configure your new SIP server.
Alternatively just search for “fwdregistrations_” on the “Configurations” form and rewrite it to your new server address.
Add the new SIP server user record at the "Users and Devices" form.
Use the Routing form and specify your custom rules for the server selection.

From SQL:
Replace OLDIP to your old SIP server IP and the NEWIP to the new SIP server IP address and then run the following SQL commands to change the upper SIP server address (copy-paste to the “Direct Query” form and hit the “Go” button):

```
update tb_users set hwid = REPLACE(hwid,'OLDIP','NEWIP'), ContractComment = REPLACE(ContractComment,'OLDIP','NEWIP'), ip = REPLACE(ip,'OLDIP','NEWIP'), authip = REPLACE(authip,'OLDIP','NEWIP'), domainname = REPLACE(domainname,'OLDIP','NEWIP'), proxyaddress = REPLACE(proxyaddress,'OLDIP','NEWIP'), transip = REPLACE(transip,'OLDIP','NEWIP'), name = REPLACE(name,'OLDIP','NEWIP'), username = REPLACE(username,'OLDIP','NEWIP')

update tb_users_authip set authip = REPLACE(authip,'OLDIP','NEWIP')

update tb_settings set valstr = REPLACE(valstr,'OLDIP','NEWIP')
```

If you (also) use a domain name, then replace OLDIP to your old SIP server domain and the NEWIP to the new SIP server domain name and then run the SQL statements again.

Using MRTC with multiple SIP server at the same time:

The WebRTC-SIP gateway can be also used with multiple SIP servers. To be able to route the users to their respective servers (register, call, chat and other sessions) you have the following possibilities:

1. Specify the upper server from your WebRTC client instead (as the SIP server domain for your app config).
   To allow client driven upper server selection, set the allowupperserverselection global config option is set to true.
2. Specify the upper server from the gateway/server routing rules:
   Add the new SIP server to the "Users and Devices"
   Use the Routing form and specify your custom rules for the server selection.
   For the routing rules to be applied also for register requests, you need to set the routingforregister global config option is set to 1.
3. Both:
   You can use both routing rules and upper server suggestions from your SIP clients.
   If the allowupperserverselection settings is true, then the client suggestion will overwrite your routing rules.

For all cases, make sure to create the necessary records from the "Users and devices" form for all your SIP servers:

- Add them as "SIP Server" (for outbound) and as "Traffic Senders" (for inbound, usually with IP authentication).
- Then configure them in the "Routing" after your needs to specify what kind of users/calls have to be routed to which SIP server.
  For example you can just list them under the "default" pattern with equal priority for load balancing or configure any rules after your requirements.

Advanced settings

WebRTC and gateway related settings are listed below for informational purposes only.
Normally these values should be left with their default values and you just need to follow the configuration wizard for a proper setup.

WebRTC module:

- haswebrtc: 1
- fs_use: 1
- haswebsocket: 1
- hasturn: 1
- localdomain: set to a valid domain name with its A record set to the server IP address

- autotlscert: 1
- usetlsforall: true
- Optional settings:
    - hasflash: 1
    - fs_pbx: 1
    - enableconferencerooms: 1
    - v2mode: 2
    - usemainaport: true
    - usetlsforweb: true
    - sslportmain: 443
    - fs_restrictsiptolocal: 1
    - fs_checkdomainmismatch: 1

Gateway mode:

- forwardauthentifications: 1
- autocreatereguser: 1
- fwdregistrations: 2
- forwardauthpassword: 1

- Optional:
    - hasbilling: 0
    - blocknotbilledcalls: 0
    - fwdunknownheaders: 2
    - fwdregistrations_xxx
    - allowupperserverselection: 1
    - runwebportal: 0
    - fastauth: 0
    - enforcestrongauth: false
    - MINUSERNAMELEN: 2
    - minpwdlength: 2
    - strongdigestauth: 0
    - allowanonymouscaller: true
    - blocksatellitecalls: false
    - blockpremiumnumbers: 0

- builddynamicblacklist: 0
- anumberhandling: 0
- enablejsonp: 2
- maxreroute: 0
- normalizenumbers: 0
- normalizedef: 0
- validateinput: 2
- normalize_clean: 1
- allowdiscmessage: false
- publicservices: 1
- dbmaint_backuplevel: 0
- dbmaint_backuptables: 0
- playadvfor: 0

How to upgrade?

To upgrade or update to the latest version, all you have to do is to run the installer. This will overwrite your old binaries while keeping all your existing data intact, so you will not lose any settings. On first startup it will automatically upgrade also the data structures, so there is no need for any manual action.

How to get support?

In case if you run into any issue, please send us the followings and we can help:

- Accurate problem description (including What you did? What should happen? What happened?) Mention also the (caller/called) username and/or SIP Call-ID or CDR ID of the problematic session(s). (The CDR ID can be found in the first column on the "CDR" form after call disconnect)
- Detailed client log (the content of your browser console) about a failed connect/register or call session (enable max log/trace level in your WebRTC client software first if it has such setting)
- If the problem is with inbound call, send the SIP Call-ID of the failed incoming call or a detailed log from your SIP server which contains also the SIP signaling (enable max log/trace level in your SIP server first if possible)
- If there are multiple issues, then send the above' s separately for each issue
- Make sure that the logs were turned on when the problem happened (or reproduce the problem with the logs enabled): Control menu -> Logs -> Set -> On

- Remote desktop (RDP) access to the machine hosting the MRTC gateway (you might need to forward the TCP and UDP 3389 if the machine if behind NAT and/or enable this port on your firewall).
  In case if you can't provide RDP access, then please send us the MRTC logs and sdf files:
  - Stop the gateway first (Control menu -> Stop server ort stop the “mserver” NT service) to release the currently opened log and sdf files.
  - The SDF files can be found in app folder: mserver.sdf (usually at C:\Program Files (x86)\MRTC\ or it can be accessed from the MRTCAdmin -> File menu -> Folders -> Work directory)
  - The log files can be found in the app Logs subfolder (can be accessed also from File menu -> Folders -> Logs directory).
    We usually need the latest file(s) or the file which contains the failed register or call sessions.
  - Compress the files and send as email attachment or if too large then make it available to download (using any file-sharing service)

Resources

- WebRTC-SIP Gateway home page
- Licensing
- Contact

Copyright © Mizutech SRL