Enterprise Guide: Microsoft Teams Integration - Metaswitch Perimeta Licensing and Configurations

Last Modified on 03/09/2023 10:50 am EST

In this article:

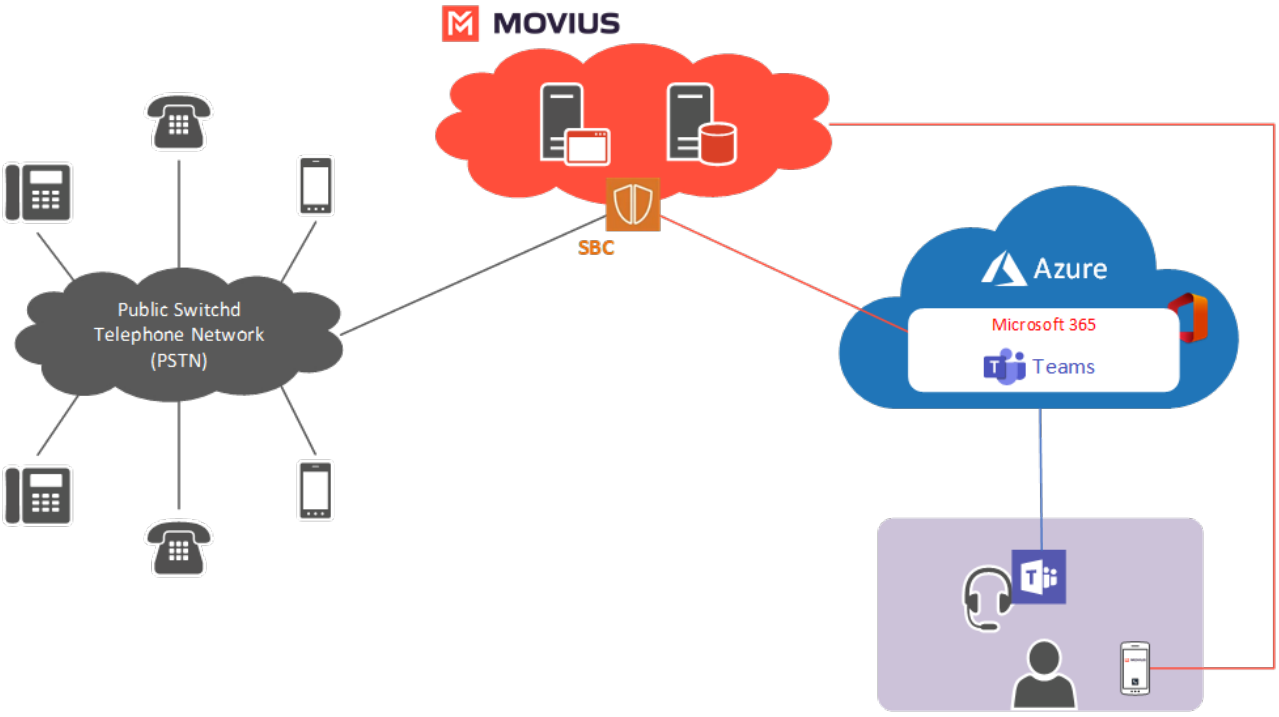
Introduction

Microsoft Teams is a unified communication and collaboration tool that provides users with voice calling, online chat, video conferencing, meetings, file storage and integration with Office 365 applications.

It is possible to provide a Microsoft Teams customer (tenant) connectivity to the PSTN by configuring a Perimeta Session Controller to interoperate with Microsoft Teams. The Session Controller connects directly to Microsoft Teams servers to provide access to the PSTN. This integration is accomplished through a trunk using the Microsoft Teams Direct Routing service.

Overview

Microsoft Phone System Direct Routing allows the connection of a supported Session Border Controller (SBC) provided by Movius to Microsoft Phone System. With this capability Movius can bring Public Switched Telephone Network (PSTN) connectivity with Microsoft Teams client in addition to the MultiLine client connectivity, as shown in the following diagram:



In this integrated scenario:

1. Microsoft Teams users can initiate external calls in the Teams application that are terminated by the Movius platform via a configured telephone service provider.
2. Make a call from any number to Movius MultiLine number and it will ring simultaneously to Microsoft Teams. Users can choose in which application the incoming calls will be answered (Movius MultiLine or Microsoft Teams).

Requirements

Licensing

Pre Requisites:

- Encryption onetime fee per deployment:
  - Signalling Encryption.
  - Media Encryption.
- Software Transcoding onetime fee per deployment:
  - SILK, G711, G722 and G729 codecs.
  - Which powers Perimeta's comfort noise interworking feature. Comfort noise interworking is required to ensure that comfort noise is present on all calls to Microsoft Teams (as required by Microsoft).

Teams Direct Routing Licensing:

- MS Teams Base License onetime fee per deployment:
  - (Media Generation and eSBC licenses was deprecated by Metaswitch for simplicity).
- Capacity License per session (per 1K):
  - Covers Non Media Bypass and Media Bypass functionality.

Prerequisites

Follow the information that is necessary to collect to configure the Perimeta SBC:

IP Configuration:

Placeholder	Example	Deployment type	Description
<session-controller-hostname>	vavsbc	All	The base FQDN of your SSC or ISC that needs to be defined. When you request a certificate later in Security certificate configuration for Microsoft Teams Direct Routing integration, you will use this FQDN as the certificate's CN (Common Name).
<microsoft-service-address-name>	GeneralAccess-01	All	The name (as configured on your SSC / ISC) of the service address your SSC / ISC uses for signaling when connecting to Microsoft Teams Direct Routing. In a geographically redundant deployment, this name must be the same on all SSC(s) / ISC(s).
<microsoft-signaling-local-port>	5058	All	The port on your SSC / ISC to use for signaling to and from Microsoft Teams Direct Routing.
<microsoft-media-ip-address>	169.57.15.98	All	The IP address your MSC / ISC uses for media sent to and from Microsoft Teams Direct Routing. You can use multiple media addresses. You can use existing media address(es).
<microsoft-media-first-port>	16384	All	The first port on your MSC(s) / ISC for media sent to and from Microsoft Teams Direct Routing. This must be an even-numbered port. Default: 16384.

<microsoft-media-last-port>	65535	All	The last port on your MSC(s) / ISC for media sent to and from Microsoft Teams Direct Routing. This must be an odd-numbered port. Default: 65535.
<microsoft-media-realm>	GeneralAccessMedia1	All	A name to identify the (group of) IP address(es) your MSC(s) / ISC(s) uses for media sent to and from Microsoft Teams Direct Routing (<microsoft-media-ip-address>). This may be an existing media realm. If you will use IP address(es) dedicated to Microsoft connectivity, Metaswitch recommends Microsoft Media. In a geographically redundant deployment, this name must be the same on all MSC(s) / ISC(s).
<pstn-or-switch-local-address-ip-version>	ipv4	All	The IP version of the service address the Session Controller uses to connect to the PSTN or your softswitch. This must be ipv4 or ipv6. If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).
<pstn-or-switch-local-address>	169.57.15.98	All	The local service address the Session Controller uses to connect to the PSTN or your softswitch, in IPv4 or IPv6 format. If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).
<pstn-or-switch-service-address-name>	GeneralAccess-01	All	The name of the service address your SSC / ISC uses for signaling when connecting to the PSTN or your softswitch. In a geographically redundant deployment, this name must be the same on all SSCs(s) / ISC(s). If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).

<pstn-or-switch-ip-version>	ipv4	All	The IP version of the range of remote IP addresses that can contact your Session Controller from the PSTN or your softswitch. If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).
<pstn-or-switch-media-ip-address>	169.57.15.98	All	The IP address your MSC(s) / ISC uses for media sent to and from the PSTN or your softswitch. You can configure multiple media addresses. You can use existing media address(es). If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).
<pstn-or-switch-media-first-port>	16384	All	The first port on your MSC(s) / ISC for media sent to and from the PSTN or your softswitch. This must be an even-numbered port. Default: 16384. If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).
<pstn-or-switch-media-last-port>	65535	All	The last port on your MSC(s) / ISC for media sent to and from the PSTN or your softswitch. This must be an odd-numbered port. Default: 65535. If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).

<pstn-or-switch-media-realm>	GeneralAccesMedia1	All	A name to identify the (group of) IP address(es) your MSC(s) / ISC use(s) for media sent to and from the PSTN (<pstn-or-switch-media-ip-address>). This may be an existing media realm. If you will use IP address(es) dedicated to this connection, Metaswitch recommends PSTN. In a geographically redundant deployment, this name must be the same on all MSC(s) / ISC(s). If you want to use different IP addresses and ports for the PSTN and for your softswitch, define two values (<switch-*> and <pstn-*>).

Session Controller Configuration:

Placeholder	Example	Deployment type	Description
<microsoft-service-network-id>	2	All	ID (an integer) of the service network you will use to communicate with Microsoft Teams Direct Routing.
<pstn-or-switch-service-network-id>	2	All	The number of the service network your MSC / ISC uses when connecting to the PSTN or your softswitch.
<microsoft-certificate-name>	vavsbc	All	A name for the security certificate your Session Controller uses to establish TLS connections to Microsoft Teams Direct Routing. You will configure this certificate as part of Security certificate configuration for Microsoft Teams Direct Routing integration.
<ip-fqdn-mapping-rule-index>	4	All	The index of the IP-FQDN mapping rule that your Session Controller will use to map the local address used to connect to your networks to <session-controller-hostname>. This must be an integer between 1 and 4,294,967,295 and must not match any other IP-FQDN mapping rule currently configured on the Session Controller.
<default-lua-config-set-index>	1	All	The index of the Lua config set that acts as the default on your Session Controller.

Configurations

There is a wide range of configuration to be set to allow the Session Controllers to integrate with Microsoft Teams. It is necessary to carry out the procedures in each of the following sections:

Security certificate configuration

It is necessary to configure the SSCs and/or ISCs with suitable local and public security certificates to encrypt and authenticate its connection to the Microsoft Teams server using TLS.

The SSCs and/or ISCs must have the following TLS certificates for interoperation with Microsoft Teams:

- A signed local security certificate that the Session Controller will use to authenticate itself to Microsoft Teams
- The certificate of the certificate authority that signed the Session Controller's local certificate (and any other certificates in the chain of trust to the root CA
- The certificate of the certificate authority that signed Microsoft Teams' certificate

Media configuration

You must configure the ISC and / or MSC(s) with suitable media realms and enable specific media features to interoperate with Microsoft Teams.

Integrating with Microsoft Teams requires specific media configuration on the ISCs and/or MSCs.

The ISCs and/or MSCs might already have some of this configuration. In all cases, it is necessary to work through this section to confirm all the required configuration and update the configuration if necessary.

- Media addresses and associated media realms for media for calls to Microsoft Teams and the PSTN
- If the firewall is configured to allow media traffic over only some of the ports that Perimeta can use for media (16384-65335), configuration limiting your ISCs and MSCs to only those ports
- At least one virtual MSC (a requirement for all media handling)
- The ability to perform transcoding and RTP/SRTP interworking
- The ability to allocate more than 5% of media bandwidth for a call to RTCP (as required by Microsoft Teams)
- Advanced media capacity (for commercial-off-the-shelf (COTS) hardware with dual CPUs with 8 or more cores without DSPs and medium- and high-capacity virtual machines with DPDK mode enabled)
- The ability to play ringback tones to transferred parties during call transfer

Example of the configuration:

config

sbc

media

media-address ipv4 **<ms-teams-media-ip-address>** service-network **<ms-teams-service-network-id>**

You can omit port-range if the default (16384-65335) is suitable

port-range **<ms-teams-media-first-port> <ms-teams-media-last-port>**

realm **<ms-teams-media-realm>**

media-address **<pstn-or-switch-ip-version> <pstn-or-switch-media-ip-address>** service-network **<pstn-or-switch-service-network-id>**

#You can omit port-range if the default (16384-65335) is suitable

port-range **<pstn-or-switch-media-first-port> <pstn-or-switch-media-last-port>**

realm **<pstn-or-switch-media-realm>**

vmsc global

activate

srtp-interworking

transcoding

increased-rtcp-bandwidth

media-playback

Note: Ringback tones for call transfer: If you want your Session Controllers to play ringback tones when calls are transferred, this feature must be enabled and ensure the media file to be played has been installed on the system.

Lua configuration

It is necessary to add a new Lua profile to the SSC or ISC to ensure that the Session Controller removes any specified RTCP ports from SDP messages, which is required to successfully interoperate with the Microsoft Teams server.

The new Lua profiles need to be added to the global default Lua configuration set on your Session Controller:

Profile name:	Remove_RTCP_a_Line
Type of SIP message body	any

Profile:	local rtcp_mux_line = msg.sdp:select_by_prefix("a=rtcp-mux")
	local rtcp_lines = msg.sdp:select_by_prefix("a=rtcp:")
	local removed_rtcp = false
	if (rtcp_mux_line[1] ~= nil) then
	for rtcp_line in rtcp_lines:iter() do
	rtcp_line:delete()
	removed_rtcp = true
	end
	if removed_rtcp then
	MeLogger.info("RTCP multiplexing is in use - remove specified RTCP port
	from SDP")
	end
	end
	---END---

Profile name:	Remove_ICE_From_SDP
Type of SIP message body:	any

Profile:	for media in MeSelection.iter(msg:get_sdp():get_media_blocks()) do
	
	for media_line in media:select_by_prefix("a=candidate"):iter() do
	
	MeLogger.info("Deleting line:".. media_line:get_text())
	
	media_line:delete()
	
	end
	
	end
	
	local ice_lines = msg.sdp:select_by_prefix("a=ice")
	
	local removed_ice = false
	
	for ice_line in ice_lines:iter() do
	
	ice_line:delete()
	
	removed_ice = true
	
	end
	
	if removed_ice then
	
	MeLogger.info("All ICE lines have been removed")
	
	end
	
	---END---

Interoperability profile and SIP MMF profile configuration

It is necessary to configure interoperability profiles and SIP Message Manipulation Framework (MMF) profiles to ensure the SSC or ISC can connect to and interoperate with the Microsoft Teams server and the PSTN.

The following configuration objects must be added to integrate the Session Controller with Microsoft Teams:

- An interoperability profile that will allow the Session Controller to successfully interoperate with the Microsoft Teams server
- A number of SIP Message Manipulation Framework (MMF) error, parameter and header profiles to ensure the Session Controller manipulates SIP messages to interoperate with the Microsoft Teams server and the PSTN or your softswitch or IMS TAS

The profiles need to be applied to the Microsoft Teams adjacencies that will be crated for Microsoft

Teams integration. Applying these profiles allows the adjacencies to interoperate correctly with the Microsoft Teams servers and the PSTN.

config

sbc

signaling

sip message-manipulation

error-profile Reject_SRTP_With_488

cause ac-srtp-disallowed status-code 48

header-profile Contact_Add_FQDN

header Contact

action modify-value sip-uri host ${adj.lcl_id}

header-profile E164_From_Match_RURI

description "Make From be E.164 if and only if Request-URI is E.164"

header _

description "Check if Request-URI is E.164 format"

action store-vars

condition advanced "(REGEX (msg.request_uri.value, '(.*sip:\\+.*)', ruri_is_e_164))"

header From entry 1

description "Check if From is E.164 format"

action store-vars

condition advanced "(REGEX (msg.first-header(\"From\").value, '(.*sip:\\+.*)', from_is_e_164))"

header From entry 2

description "Split up From header to allow inserting of country code if needs be"

action store-vars

condition advanced "(REGEX (msg.first-header(\"From\").value, '(.*<sip:)(\\<country-code>)\?(.*)' ,from_before_cc, _, from_after_cc))"

header From entry 3

description "Insert country code if Request-URI is E.164,and From header is not"

action replace-value value ${from_before_cc}<country-code>${from_after_cc}

condition advanced "((DEFINED (ruri_is_e_164)) AND (NOT (DEFINED (from_is_e_164))))"

header From entry 4

description "Remove country code if Request-URI is not E.164, and From is"

action replace-value value ${from_before_cc}${from_after_cc}

condition advanced "((NOT (DEFINED (ruri_is_e_164))) AND (DEFINED (from_is_e_164)))"

header-profile SubResponses

header Store1

action store-vars

condition advanced "(STORE (subdomain, msg.first-header(\"To\").uri.sip_uri.host))"

header Contact

action modify-value sip-uri host ${subdomain}

condition advanced "((DEFINED (subdomain)) AND (NOT (msg.is_request)))"

header-profile Strip_Privacy_If_Not_Anonymous

description "Remove Privacy header from Teams calls unless From is anonymous"

header Privacy

action strip

condition advanced "(NOT (REGEX (msg.first-header(\"From\").value, '.*anonymous.*')))"

header-profile Set_Contact_SubDomain

description "Convert steering prefix into Teams subdomain if set"

header To

description "Strip steering prefix from the To header if found"

action modify-value sip-uri user ${original_called_party}

condition advanced "(DEFINED (steering_prefix))"

header Store

description "Detect a Teams steering prefix added to the called party user ID by routing and store the steering prefix and the original ID"

action store-vars

condition advanced "(REGEX (msg.called_party_id.user_id, 'AA(.*)AA(.*)', steering_prefix, original_called_party))"

header Contact

description "Use steering prefix to set Teams sub-domain in the Contact header"

Change to sbc${steering_prefix} if subdomains are sbc123 etc.

action modify-value sip-uri host ten${steering_prefix}.${original_contact_host}

condition advanced "(DEFINED (steering_prefix))"

header Store entry 2

description "Store the hostname from the Contact header"

action store-vars

condition advanced "(STORE (original_contact_host, msg.first-header(\"Contact\").uri.sip_uri.host))"

header Request-URI

description "Strip steering prefix from the Request-URI if found"

action modify-value sip-uri user ${original_called_party}

condition advanced "(DEFINED (steering_prefix))"

GR: add one ip-fqdn-mapping-rule for each SSC/ISC

GR: use <ssc1-pstn-switch-signaling-ip> and <ssc1-hostname> etc.

ip-fqdn-mapping-rule <ip-fqdn-mapping-rule-index> <pstn-or-switch-local-address-ip-version> <pstn-or-switch-local-address> <session-controller-hostname> both-ways

sip interop-profile Teams

header-settings contact add tls-param

header-settings from rewrite host local port exclude

header-settings to rewrite

hunting-trigger 503

hunt-on-no-1xx timeout 5000 hunt-mode standard

ping-enable

ping-mechanism pause-during-traffic

options-ping-response define-success 200-299

ping-response

respond-when-no-username always

respond-when-username always

message-manipulation

edit-profiles inbound Remove_RTCP_a_Line

edit-profiles outbound
Contact_Add_FQDN,Remove_RTCP_a_Line,E164_From_Match_RURI,SubResponses

ms-teams version-header enabled

activate

Address group configuration

It is necessary to configure IP addresses for Microsoft Teams as address groups on the SSC or ISC. You will then configure your Microsoft Teams adjacencies to allow traffic from these address groups.

The SSC or ISC must be configured to treat the IP addresses to which Microsoft Teams FQDNs resolve as known sources. The Session Controller recognizes known sources as sources of legitimate traffic but can dynamically blacklist these sources if the amount or type of traffic becomes suspicious. Address groups allow to group together non-sequential IP addresses. The groups of IP addresses can be applied to adjacencies as trusted remote address groups, which represent the sources of traffic that the adjacencies should accept. Marking the remote address group as trusted configures the group's IP addresses as known sources on that adjacency.

It is necessary to configure three address groups. The addresses in each group will be the IP addresses to which the Microsoft Teams FQDNs resolve.

- The 1st address group will contain the IP address to which ip.pstnhub.microsoft.com

resolves.

- The 2nd address group will contain the IP address to which ip2.pstnhub.microsoft.com

resolves.

- The 3rd address group will contain the remaining Microsoft Teams IP addresses.

Determining the IP addresses for each address group : The three Microsoft Teams FQDNs (**sip.pstnhub.microsoft.com**, **sip2.pstnhub.microsoft.com** and **sip3.pstnhub.microsoft.com**) will resolve to one of the following IP addresses, depending on your geographic location:

- 52.114.148.0
- 52.114.132.46
- 52.114.75.24
- 52.114.76.76
- 52.114.7.24
- 52.114.14.70
- 52.114.16.74
- 52.114.20.29
- 52.114.36.156
- 52.114.32.169

Please see:

https://docs.microsoft.com/en-us/microsoftteams/direct-routing-plan#sip-signaling-fqdns

for the most recent IP addresses.

To determine how to split the IP addresses into the address groups based on how the FQDNs resolve in your geographic region.

- Perform a DNS lookup to determine the IP address to which sip.pstnhub.microsoft.com resolves. Metaswitch recommends dig or nslookup. This will be <ms-teams-ip-1> in the example configuration
- Perform a DNS lookup to determine the IP address to which sip2.pstnhub.microsoft.com resolves. This will be <ms-teams-ip-2>
- Use the remaining IP addresses as the values of the remaining <ms-teams-ip-X> placeholders

Adjacency configuration

It is necessary to configure adjacencies on the SSC or ISC to represent the SIP peers that the Session Controller will be communicating with.

This section contains example adjacency configuration for the following connections:

- The server that your Session Controller will try first (sip.pstnhub.microsoft.com)
- A second server that your Session Controller will connect to if the primary server is not available (sip2.pstnhub.microsoft.com)
- A third server that your Session Controller will connect to if the primary and secondary servers are not available (sip3.pstnhub.microsoft.com)
- The PSTN (Telephone provider)

Example Microsoft Teams adjacency configuration:

config

sbc

signaling

adjacency sip TeamsPrimary

add-route-header sip.pstnhub.microsoft.com

call-media-policy

media-bypass-policy forbid

comfort-noise-codec interwork

secure-media require

transcoding

trigger retry-on-4xx

rtcp-multiplexing enabled

rtcp-transmission-policy always

media-playback enabled

listen-transports tcp

adjacency-type preset-peering

local-id host <session-controller-hostname> # GR: use <deploymentfqdn>

message-manipulation

error-profile outbound Reject_SRTP_With_488

lua-config-set <default-lua-config-set-index>

privacy trusted

realm <ms-teams-media-realm>

contact-username passthrough

tls fqdn sip.pstnhub.microsoft.com

certificate-name <ms-teams-certificate-name>

service-address <ms-teams-service-address-name>

signaling-local-port <ms-teams-signaling-local-port>

remote-address-group msteams-primary trusted

signaling-peer sip.pstnhub.microsoft.com

signaling-peer-port 5061

default-interop-profile Teams

activate

adjacency sip TeamsSecondary

add-route-header sip2.pstnhub.microsoft.com

call-media-policy

media-bypass-policy forbid

comfort-noise-codec interwork

secure-media require

transcoding

trigger retry-on-4xx

rtcp-multiplexing enabled

rtcp-transmission-policy always

media-playback enabled

listen-transports tcp

adjacency-type preset-peering

local-id host <session-controller-hostname> # GR: <deployment-fqdn>

message-manipulation

error-profile outbound Reject_SRTP_With_488

lua-config-set <default-lua-config-set-index>

privacy trusted

realm <ms-teams-media-realm>

contact-username passthrough

tls fqdn sip.pstnhub.microsoft.com

certificate-name <ms-teams-certificate-name>

service-address <ms-teams-service-address-name>

signaling-local-port <ms-teams-signaling-local-port>

remote-address-group msteams-secondary trusted

signaling-peer sip2.pstnhub.microsoft.com

signaling-peer-port 5061

default-interop-profile Teams

activate

adjacency sip TeamsTertiary

add-route-header sip3.pstnhub.microsoft.com

call-media-policy

media-bypass-policy forbid

comfort-noise-codec interwork

secure-media require

transcoding

trigger retry-on-4xx

rtcp-multiplexing enabled

rtcp-transmission-policy always

media-playback enabled

listen-transports tcp

adjacency-type preset-peering

local-id host <session-controller-hostname> # GR: use <deploymentfqdn>

message-manipulation

error-profile outbound Reject_SRTP_With_488

lua-config-set <default-lua-config-set-index>

privacy trusted

realm <ms-teams-media-realm>

contact-username passthrough

tls fqdn sip.pstnhub.microsoft.com

certificate-name <ms-teams-certificate-name>

service-address <ms-teams-service-address-name>

signaling-local-port <ms-teams-signaling-local-port>

remote-address-group msteams-tertiary trusted

signaling-peer sip3.pstnhub.microsoft.com

signaling-peer-port 5061

default-interop-profile Teams

activate

Example PSTN adjacency configuration

config

sbc

signaling

adjacency sip PSTN

call-media-policy

media-bypass-policy forbid

comfort-noise-codec passthrough

transcoding

trigger retry-on-4xx

rtcp-multiplexing enabled

media-playback enabled

interop

header-settings from rewrite host local port exclude

header-settings to rewrite

message-manipulation

edit-profiles inbound Remove_RTCP_a_Line

edit-profiles outbound Remove_RTCP_a_Line,Remove_ICE_From_SDP, Strip_Privacy_If_Not_Anonymous, E164_From_Match_RURI

force-signaling-peer initial-requests

adjacency-type preset-peering

message-manipulation

error-profile outbound Reject_SRTP_With_488

lua-config-set <default-lua-config-set-index>

privacy trusted

realm <pstn-or-switch-media-realm>

service-address <pstn-or-switch-service-address-name>

remote-address-range <pstn-or-switch-ip-version> <pstn-first-remote-address> prefix-len <pstn-prefix-length> trusted

signaling-peer <pstn-first-remote-address>

signaling

activate

Routing configuration

It is necessary to update the call policy set on the SSC or ISC to ensure that the Session Controller can route calls successfully between the configured adjacencies.

It is necessary to configure routing tables that allow the Session Controller to do the following:

- For a call from Microsoft Teams: route the call to the Movius platform adjacency
- For a call from the Movius platform: route the call to a Microsoft Teams adjacency

Metaswitch recommendation is to use the following types of table. These tables might be the only table on the Session Controller, or they might be integrated into a more complex call policy set:

- A source adjacency routing table to detect whether a call is from Microsoft Teams or the Movius platform and select an adjacency or further routing tables
- A least-cost routing table to select the Microsoft Teams adjacency to use
- To use Microsoft Teams call transfer features, it is necessary to add an initial destination domain routing table which routes the base FQDN for your SSC or ISC back to MS teams

Source adjacency table

Configure a source adjacency table that all calls to or from Microsoft Teams will reach. If you do not already have an existing call policy set, this will be the first table for new call requests:

- The first entry matches requests arriving on the Movius platform adjacency and sets the next table to the least cost routing table for selecting Microsoft Teams adjacencies
- Three further entries route requests arriving on the TeamsPrimary, TeamsSecondary and TeamsTertiary adjacencies over the Movius platform adjacency and allow the Session Controller to complete routing for these calls

rtg-src-adjacency-table msteams

entry 1

match-adjacency PSTN

action next-table steering

entry 2

match-adjacency TeamsPrimary

dst-adjacency PSTN

action complete

entry 3

match-adjacency TeamsSecondary

dst-adjacency PSTN

action complete

entry 4

match-adjacency TeamsTertiary

dst-adjacency PSTN

action complete

Least cost routing table

Configure a least cost routing table that routes calls that have passed through the destination ID table out over a Microsoft Teams adjacency. This routing table has three entries:

- The first entry routes over the TeamsPrimary adjacency
- If the connection to the primary Microsoft Teams FQDN has failed (making the TeamsPrimary adjacency unavailable), the second routes requests over the TeamsSecondary adjacency
- If the connections to the primary and secondary Microsoft Teams FQDNs have failed, the third entry routes requests over the TeamsTertiary adjacency

rtg-least-cost-table teams-redundancy

entry 1

cost 1

dst-adjacency TeamsPrimary

action complete

entry 2

cost 2

dst-adjacency TeamsSecondary

action complete

entry 3

cost 3

dst-adjacency TeamsTertiary

action complete

References

Microsoft Teams and Perimeta Integration Guide

https://manuals.metaswitch.com/Perimeta/V4.8/MicrosoftTeamsIntegrationGuide/Source/notices.html

Movius vavMeta Configuration

Interoperability requirements

Summary of required signaling features for interoperation with Microsoft Teams:

Area of configuration	Connections to Microsoft Teams	Connections to the PSTN
Permitted types of connection	Send outbound traffic to a specified Microsoft Teams FQDN. Accept inbound traffic from specific IP addresses (configured as a remote address group). Use TLS and do not accept incoming UDP connections.	No specific configuration required.
Peer detection with SIP OPTIONS requests	Respond to all SIP OPTIONS requests (with or without usernames) (ping-response ... on interoperability profile). Use requests to detect the availability of peers, pausing during traffic (ping-enable ... on interoperability profile). Treat any responses to OPTIONS requests other than 200-299 as failures (options-ping-response ... on interoperability profile).	Recommended but not required and not included in this manual. For instructions, see *Configuring peer availability detection for a static signaling peer* in *Perimeta Configuration and Interoperability Guide - CLI Users*.
Routing of calls	Hunt on receiving a 503 response and when the Session Controller does not receive a 1XX response after 5 seconds (hunting-trigger ... and hunt-on-no-1xx ... on interoperability profile).	Connect to the PSTN peer and force all outbound out-of-dialog and dialog-creating requests on this adjacency to it.

Rewriting outbound Contact headers	Add the IP address of this adjacency to the Contact header (edit-profiles outbound Contact_Add_FQDN on interoperability profile). Add a transport=tls parameter (header-settings contact ... on interoperability profile). Pass through usernames in Contact headers (contact-username ...).	No specific configuration required.
Rewriting outbound From headers	Use the Session Controller's FQDN as the hostname and exclude the local port (header-settings from ... on interoperability profile and local-id host ...).	Use the Session Controller's signaling IP address on this adjacency as the hostname and exclude the local port (header-settings from ...)
Rewriting outbound To headers	Rewrite to contain the outbound Request-URI (header-settings to ... on interoperability profile).	Rewrite to contain the outbound Request-URI (header-settings to ...)
Rewriting inbound messages with message manipulation rules	Remove a=rtcp lines from SDP (edit-profiles inbound on interoperability profile). Convert 183 (SDP) responses sent by Microsoft Teams to 180.	Remove a=rtcp lines from SDP (edit-profiles inbound ..-)

Rewriting other parts of outbound messages with message manipulation rules	Remove a=rtcp lines from SDP. Convert the From header to the E.164 format if the Request-URI is in E.164. Set the Contact header to include the per-tenant subdomain of the Session Controller. (edit-profiles outbound ... on interoperability profile).	Remove a=rtcp lines from SDP Remove a=candidate lines (for interactive connectivity checks) from SDP Remove Privacy headers if the From header does not contain anonymous. Convert the From header to the E.164 format if the Request-URI is in E.164 (edit-profiles outbound ...)
SIP privacy	Do not remove user identifiers (privacy trusted).	Do not remove user identifiers (privacy trusted).
Adding an X-MS-SBC header to messages. This header specifies the SBC vendor, model and version on all outbound SIP messages in calls and SIP OPTIONS requests.	Required by Microsoft Teams. (ms-teams ... on interoperability profile)	No specific configuration required. This header is not present on messages sent or received on this adjacency.
Microsoft Teams call transfer*	Can be enabled with transfer-policy > ms-teams-transfer on adjacency. Also requires media-playback enabled.	For versions earlier than V4.8.20, secure-media-forbid required.

Summary of required media features for interoperation with Microsoft Teams

Area of configuration	Connections to Microsoft Teams	Connections to the PSTN

Comfort noise	Interwork to provide comfort noise (comfort-noise-codec interwork) as required by Microsoft Teams.	Pass through any existing comfort noise (comfort-noise-codec passthrough).
Codecs and transcoding	Codecs supported by Microsoft Teams (all supported by Perimeta automatically). SILK G.711 (PCMU and PCMA) G.722 G.729 Transcoding is required to support comfort noise interworking.	Transcode on receiving a 4XX response from an endpoint (transcoding ...). You may need to restrict codecs on this side of the call, depending on your network and endpoints. Refer to the *Hardware and software transcoding* section of *Media transcoding* in *Perimeta Configuration and Interoperability Guide - CLI Users* to determine which codecs are not supported by your system and ensure that they are restricted using a suitable codec list. For more information on codec lists, see *Codec lists* in *Perimeta Configuration and Interoperability Guide - CLI Users*.
RTCP	Use RTCP multiplexing (rtcp-multiplexing enabled). Ensure dialog always has RTCP (rtcp-transmission-policy always). Microsoft Teams requires RTCP on all calls and RTCP multiplexing.	Use RTCP multiplexing (rtcp-multiplexing enabled).
SRTP	Require SRTP (secure-media require) Use 488 as the error code if the call is rejected because SRTP is not allowed (error-profile outbound Reject_SRTP...)	Use 488 as the error code if the call is rejected because SRTP is not allowed (error-profile outbound Reject_SRTP...).

Media bypass	Forbid (media-bypass forbid). Required to allow Perimeta to provide comfort noise interworking.	Forbid (media-bypass forbid). Required to permit transcoding.
DTMF	Microsoft Teams does not support inband DTMF tones. It requires RFC 2833 DTMF.	No specific configuration required, unless endpoints use inband tones to signal DTMF. If endpoints do use inband DTMF, you must configure the Session Controller to perform inband tones interworking.
Call hold interworking	Deployments with a MetaSphere CFS: Use c=IN IP4 0.0.0.0 to signal call hold towards Microsoft Teams (hold-setting...) When the CFS has placed a call on hold, instruct Microsoft Teams to send media to a Perimeta media address used for Microsoft Teams. Perimeta will drop the media while the call is on hold (edit-profiles outbound Drop_Media...)	No specific configuration required.
Playing a media file for Microsoft Teams call transfer	Enable media playback on adjacency (media-playback enabled on adjacency) Install media file to play on ISCs and MSCs, with ID ringback. Enable media playback and RTP/SRTP interworking on ISCs and MSCs (media-playback and srtp-interworking)	No specific configuration required.

MOVIUS