# Overview on CMS presentation sharing with Skype for Business using Expressway-E as TURN server - Cisco

## Contents



## Introduction

This document describes a detailed view on the TCP TURN message exchange between the CMS, Expressway and Skype for Business components.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Expressway server
- CMS (Cisco Meeting Server)
- Skype for Business (previously Lync) server

### Components Used

The information in this document is based on these software and hardware versions:

- Expressway 8.9

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

# Background Information

Expressway version X8.9 introduced support for TCP TURN, allowing presentation sharing calls between CMS and Skype for Business (Lync) where CMS would use Expressway-E as its TURN server. Content media from the Skype client is then expected to flow towards Expressway-E, which then forwards it to CMS on premise.

This document is supposed to provide a detailed view on the TCP TURN message exchange between all the components to help troubleshoot the potential issues. It does not explain the fundamentals of TURN or the use of UDP TURN for regular audio or video call.

> **Tip**: The TCP TURN is an extension to TURN documented under the following RFC6062.

This document focuses on the TCP part, which is unique for Skype presentation sharing calls, and adds extra complexity to the classic TURN operation.

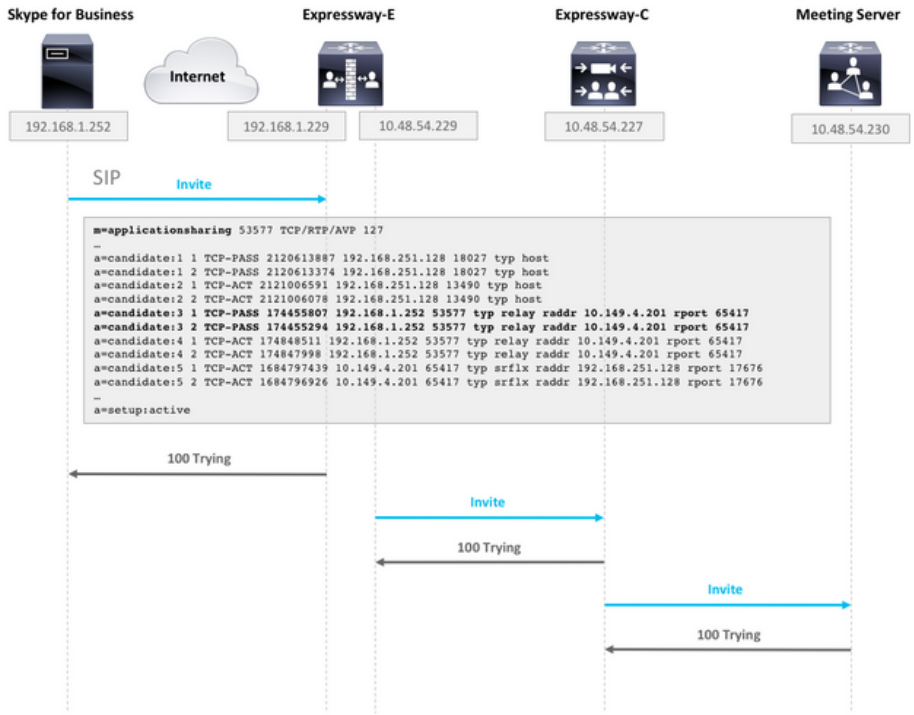
# Scenario

In the test lab scenario described in this document, we have Skype client communicating to CMS over Skype Edge server, Expressway-E and Expressway-C. Expressway-E is configured in CMS as a TURN server. Additionally, the Skype client has no IP connectivity to the Expressway-E server, so we expect the only working media path to be over the Skype Edge towards the Expressway-E server.

## Network Diagram

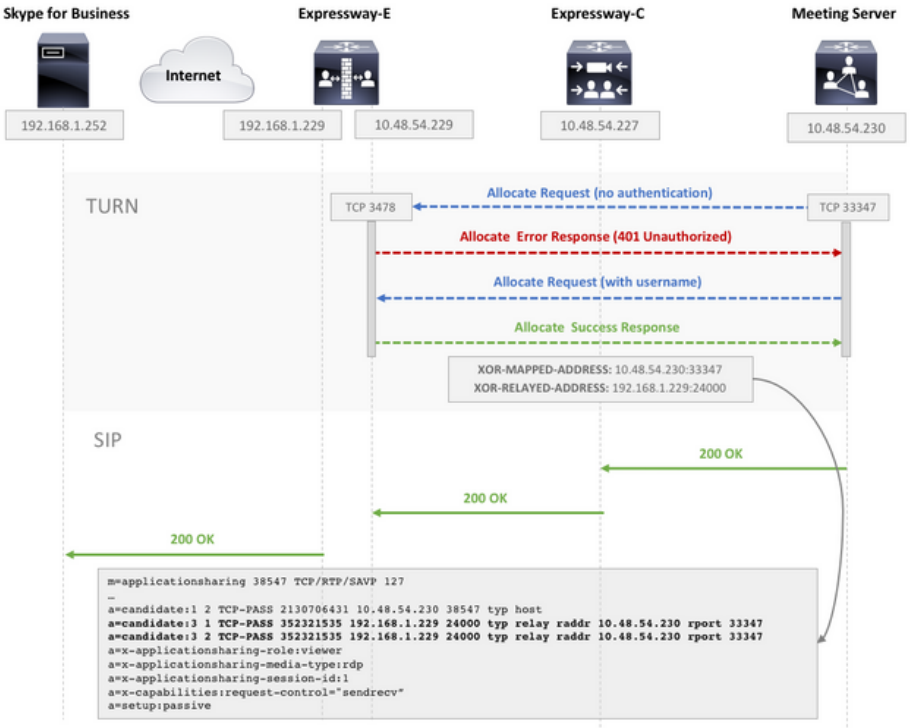
The following image shows the new **INVITE** with **m=applicationsharing** is sent from Skype to initiate the presentation sharing.

(it does not show the initial audio and video call invites, which are already negotiated at this stage):




SDP from Skype contains remote ICE candidates. Note the m=applicationsharing that indicates this is a call for sharing presentation. It will have a different SIP call-id than the initial audio/video call.

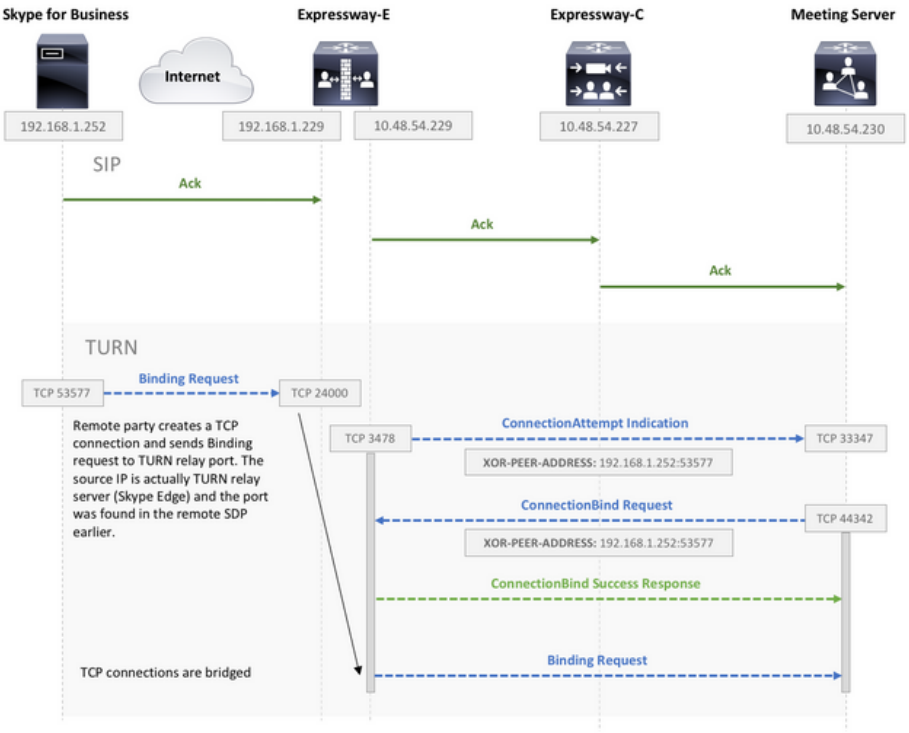
After CMS receives the call, it will reach out to its TURN server (Expressway-E) to get its own TURN relay candidates.




CMS make TCP connection to TURN server for TURN relay candidate allocation.

TURN server sends Allocate Success Response which contains the TURN relay candidate.

CMS adds TURN relay candidate to SDP in its 200 OK SIP response.



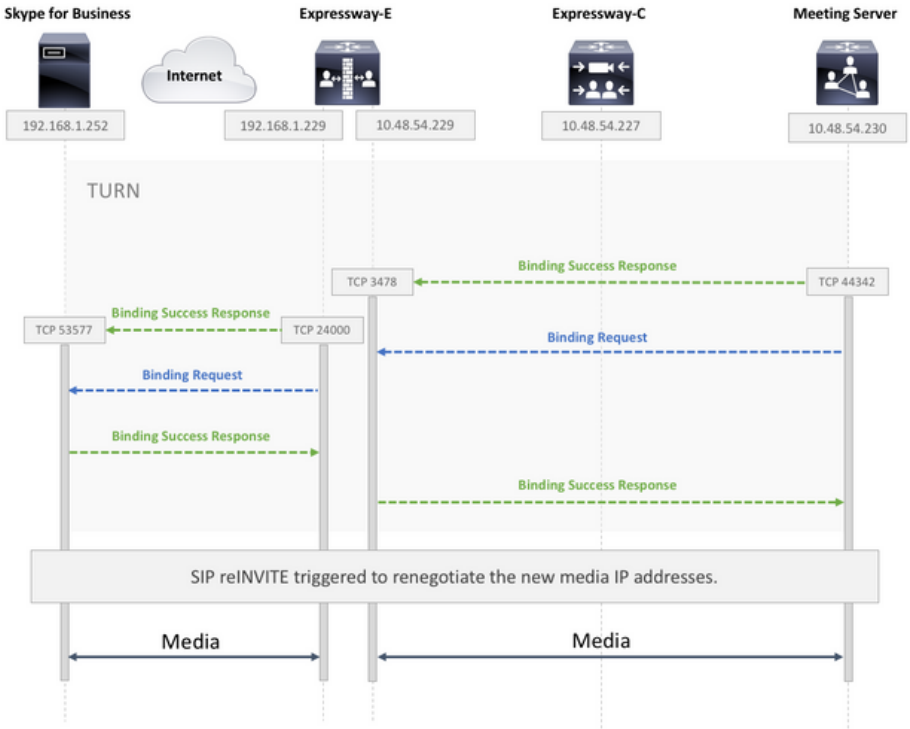

SIP dialog finishes with the ACK

TURN server notifies the TURN client about a connection made to the relay candidate address (XOR-PEER-ADDRESS attribute). This is done over the same TCP connection where Allocate Request was sent.

TURN client (CMS) creates a new TCP connection to TURN server to request the remote XOR-PEER-ADDRESS connection to be bridged to this new TCP connection.

TURN server confirms. From now on the traffic from remote peer 192.168.1.252:53577 hitting port 24000 on TURN server will be forwarded over this TCP connection to CMS.

Binding request from Skype is sent to CMS.




Bidirectional Binding Requests and Binding Success Responses are required for this candidate pair to be considered valid.

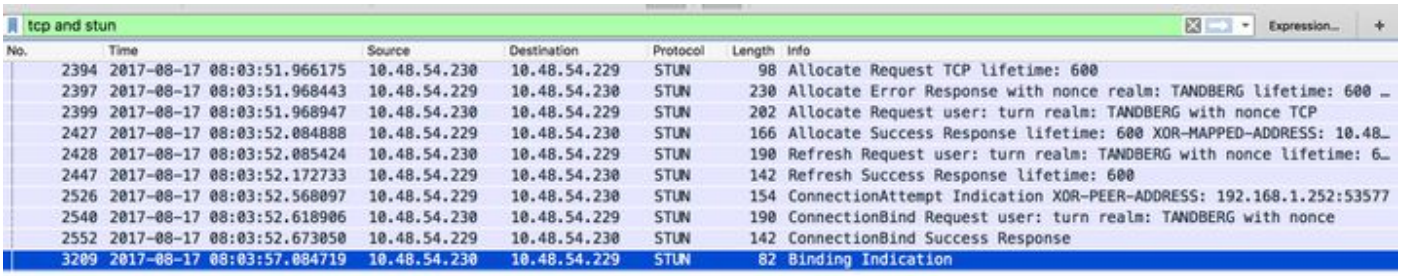
After Binding Success Response was received in both directions, there will be SIP reINVITE dialog between CMS and Skype to establish the new media route.

# Working with packet captures

## Wireshark filter

In some situations, in order to get quick overview of the STUN communication, it may be enough

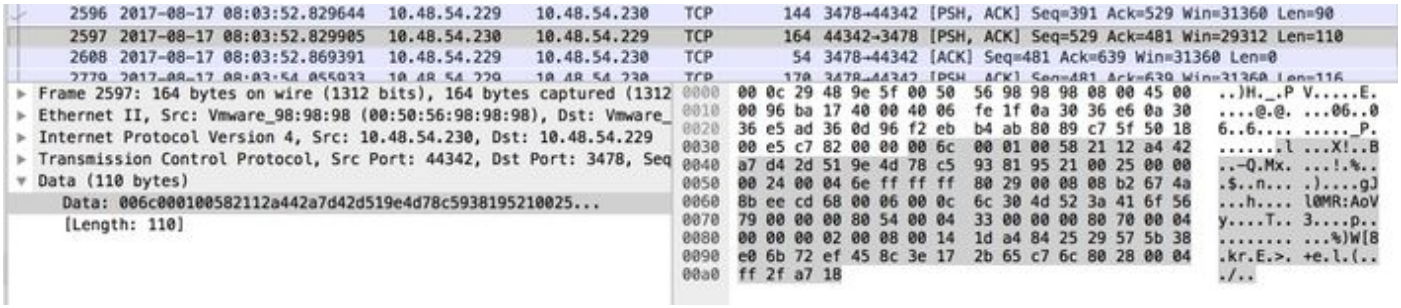
to set a Wireshark filter as **tcp and stun:**




## Looking for STUN packets in TCP payload

Wireshark may not always decode the TCP communication as STUN.

You will have to filter out on the TCP port which is used for communication, look for TCP packets with **[PSH, ACK]** flag and investigate the TCP payload:




In the the image above the payload begins with data **00 6c 00 01**. The different values in the 3rd and 4th byte represent the following STUN packets:

00 01 - Binding Request

01 01 - Binding Success Response

In order for the STUN pair to work, there has to be one of each in each direction.

## Using Wireshark to decode MSSTUN messages

Microsoft has made additions to the base IETF standards which are not recognised by Wireshark. You can install a plugin into Wireshark which will make these packet capture more readable.

More information on the plugin can be found here.

# Troubleshoot

This section provides information you can use in order  to troubleshoot your configuration.

## User is not able to share

- Check if the CMS logs contain the following entry: **ms-diagnostics-public: 21002;reason="Attendees cannot share in this conference";component="ASMCU"**
- Skype for Business Meetings are not setup to allow all to share by default. If you see the

above error, right click on the attendee from the Skype client and select **Make Presenter**