[MS-ICE2BWM]:

Interactive Connectivity Establishment (ICE) 2.0 Bandwidth Management Extensions

---

technology. Because this documentation might change between the pre-release version and the final version of this technology, there are risks in relying on this preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
3/31/2010	0.1	Major	Initial Availability
4/30/2010	0.2	Editorial	Revised and edited the technical content
6/7/2010	0.3	Editorial	Revised and edited the technical content
6/29/2010	0.4	Editorial	Changed language and formatting in the technical content.
7/23/2010	0.4	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.0	Major	Significantly changed the technical content.
11/15/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.0	Major	Significantly changed the technical content.
4/11/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.0.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	2.0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.1	Minor	Clarified the meaning of the technical content.
11/18/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.2	Minor	Clarified the meaning of the technical content.
7/31/2014	2.3	Minor	Clarified the meaning of the technical content.
10/30/2014	2.3	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	3.0	Major	Significantly changed the technical content.
9/4/2015	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/15/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
9/14/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
4/27/2018	4.0	Major	Significantly changed the technical content.
7/24/2018	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the **Interactive Connectivity Establishment (ICE)** 2.0 Bandwidth Management Extensions. This protocol consists of a set of proprietary extensions to the Interactive Connectivity Establishment (ICE) Extensions 2.0, as described in [MS-ICE2].

The protocol described in [MS-ICE2] specifies how to set up **Real-Time Transport Protocol (RTP)** streams in a way that allows the streams to traverse **network address translation (NAT)** and firewalls. The protocol described in [MS-ICE2] is agnostic to bandwidth or other policy constraints and attempts to find the highest priority path for the media session.

This protocol specifies how to determine and enforce bandwidth policy constraints by communicating with a bandwidth policy aware server.

This protocol facilitates:

- Communication with a server based on the protocol described in [MS-TURNBWM] that supports network bandwidth utilization management and access control. The server is known as a bandwidth policy server. The bandwidth policy server uses this protocol to determine any policy constraints that necessitate avoiding viable media paths that could potentially be used for media flow.
- Enforces bandwidth policy constraints and ensures that policy restricted paths are not used for media flow.
- Periodically reports to the bandwidth policy server the path and the bandwidth being utilized by the media session.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

agent: A device that is connected to a computer network. Also referred to as an endpoint.

answer: A message that is sent in response to an **offer** that is received from an offerer.

bandwidth management endpoint: A protocol client that communicates with a protocol server to discover and enforce applicable bandwidth policies, and to track and send updates about bandwidth utilization to that server.

callee: An **endpoint** to which a call is initiated by a **caller**.

caller: An **endpoint** that initiates a call to establish a media session.

candidate: A set of **transport addresses** that form an atomic unit for use with a media session. For example, in the case of Real-Time Transport Protocol (RTP) there are two transport addresses for each candidate, one for RTP and another for the Real-Time Transport Control Protocol (RTCP). A candidate has properties such as type, priority, foundation, and base.

candidate pair: A set of candidates that is formed from a **local candidate** and a **remote candidate**.

Check List: An ordered list of **candidate pairs** that determines the order in which connectivity checks are performed for those candidate pairs.

component: A representation of a constituent **transport address** if a **candidate** consists of a set of transport addresses. For example, media streams that are based on the Real-Time Transfer

Protocol (RTP) have two components, one for RTP and another for the Real-Time Transfer Control Protocol (RTCP).

connectivity check: A **Simple Traversal of UDP through NAT (STUN)** binding request that is sent to validate connectivity between the local and remote candidates in a **candidate pair**.

controlled agent: An Interactive Connectivity Establishment (ICE) agent that waits for the controlling agent to select the final **candidate pairs** to be used.

controlling agent: An **Interactive Connectivity Establishment (ICE)** agent that is responsible for selecting and signaling the final **candidate pair** that is selected by connectivity checks. The controlling agent signals the final **candidates** in a **Simple Traversal of UDP through NAT (STUN)** binding request and an updated offer. In a session, one of the agents is a controlling agent and the other agent is a controlled agent.

endpoint: A device that is connected to a computer network.

final offer: An offer that is sent by a **caller** at the end of connectivity checks and carries the **local candidate** and the **remote candidate** that were selected for media flow.

full: An Interactive Connectivity Establishment (ICE) implementation that adheres to the complete set of functionality described in [MS-ICE2].

Host Candidate: A **candidate** that is obtained by binding to ports on the local interfaces of the host computer. The local interfaces include both physical interfaces and logical interfaces such as Virtual Private Networks (VPNs).

initial offer: An offer that is sent by a **caller** and with the caller's **local candidates** when the caller initiates a media session with a **callee**.

Interactive Connectivity Establishment (ICE): A methodology that was established by the Internet Engineering Task Force (IETF) to facilitate the traversal of network address translation (NAT) by media.

Internet Protocol version 4 (IPv4): An Internet protocol that has 32-bit source and destination addresses. IPv4 is the predecessor of IPv6.

Internet Protocol version 6 (IPv6): A revised version of the Internet Protocol (IP) designed to address growth on the Internet. Improvements include a 128-bit IP address size, expanded routing capabilities, and support for authentication and privacy.

INVITE: A **Session Initiation Protocol (SIP)** method that is used to invite a user or a service to participate in a session.

Lite: An implementation that supports a minimal subset of Interactive Connectivity Establishment (ICE) functionality, as described in [MS-ICE2], to work with a Full ICE implementation. A Lite implementation responds to but does not send connectivity checks.

local candidate: A **candidate** whose transport addresses are local transport addresses.

local transport address: A transport address that is obtained by binding to a specific port from an IP address on the host computer. The IP address can be from physical interfaces or from logical interfaces such as Virtual Private Networks (VPNs).

network address translation (NAT): The process of converting between IP addresses used within an intranet, or other private network, and Internet IP addresses.

offer: A message that is sent by an offerer.

peer: An additional **endpoint** that is associated with an endpoint in a session. An example of a peer is the **callee** endpoint for a **caller** endpoint.

peer-derived candidate: A **candidate** whose **transport addresses** are new mapping addresses, typically allocated by **NATs**, that are discovered during **connectivity checks**.

provisional answer: An optional message that carries **local candidates** for a **callee** and can be sent by the callee in response to a **caller's** initial offer.

Real-Time Transport Control Protocol (RTCP): A network transport protocol that enables monitoring of Real-Time Transport Protocol (RTP) data delivery and provides minimal control and identification functionality, as described in [RFC3550].

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

Relayed Candidate: A **candidate** that is allocated on the Traversal Using Relay NAT (TURN) server by sending an Allocate Request to the TURN server.

remote candidate: A **candidate** that belongs to a remote **endpoint** in a session.

remote endpoint: See **peer**.

Server Reflexive Candidate: A **candidate** whose transport addresses is a **network address translation (NAT)** binding that is allocated on a NAT when an **endpoint** sends a packet through the NAT to the server. A Server Reflexive Candidate can be discovered by sending an allocate request to the **TURN server** or by sending a binding request to a **Simple Traversal of UDP through NAT (STUN)** server.

Session Description Protocol (SDP): A protocol that is used for session announcement, session invitation, and other forms of multimedia session initiation. For more information see [MS-SDP] and [RFC3264].

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

Simple Traversal of UDP through NAT (STUN): A protocol that enables applications to discover the presence of and types of network address translations (NATs) and firewalls that exist between those applications and the Internet.

STUN candidate: A **candidate** whose transport addresses are STUN-derived transport addresses. See also **Simple Traversal of UDP through NAT (STUN)**.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

transport address: A 3-tuple that consists of a port, an IPv4 or IPV6 address, and a transport protocol of User Datagram Protocol (UDP) or Transmission Control Protocol (TCP).

Traversal Using Relay NAT (TURN): A protocol that is used to allocate a public IP address and port on a globally reachable server for the purpose of relaying media from one **endpoint** to another **endpoint**.

TURN candidate: A **candidate** whose transport addresses are TURN-derived transport addresses. See also **Traversal Using Relay NAT (TURN)**.

TURN server: An **endpoint** that receives **Traversal Using Relay NAT (TURN)** request messages and sends TURN response messages. The protocol server acts as a data relay, receiving data on the public address that is allocated to a protocol client and forwarding that data to the client.

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-ICE2] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions 2.0".

[MS-TURNBWM] Microsoft Corporation, "Traversal using Relay NAT (TURN) Bandwidth Management Extensions".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

1.2.2 Informative References

[IETFDRAFT-ICENAT-19] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19, October 2007, http://tools.ietf.org/html/draft-ietf-mmusic-ice-19

[MS-TURN] Microsoft Corporation, "Traversal Using Relay NAT (TURN) Extensions".

1.3 Overview

Managing and controlling utilization of network bandwidth is important for enterprises to reduce cost and also to ensure good quality of service. Media communication traffic has the potential to congest or overuse the available bandwidth on network links unless the utilization is closely monitored and bandwidth policy restrictions are actively enforced. Even if the bandwidth utilization is known, enforcing the bandwidth policy is difficult because the clients involved in the media session could be dispersed across the enterprise and can be in different network or geographical regions.

The Interactive Connectivity Establishment (ICE) Extensions 2.0, as described in [MS-ICE2], are used to establish media flow between a **caller endpoint** and a **callee** endpoint. This protocol seamlessly integrates with and extends the protocol described in [MS-ICE2] for bandwidth management.

The following figure shows a typical deployment scenario with two endpoints that establish a media session with bandwidth management.

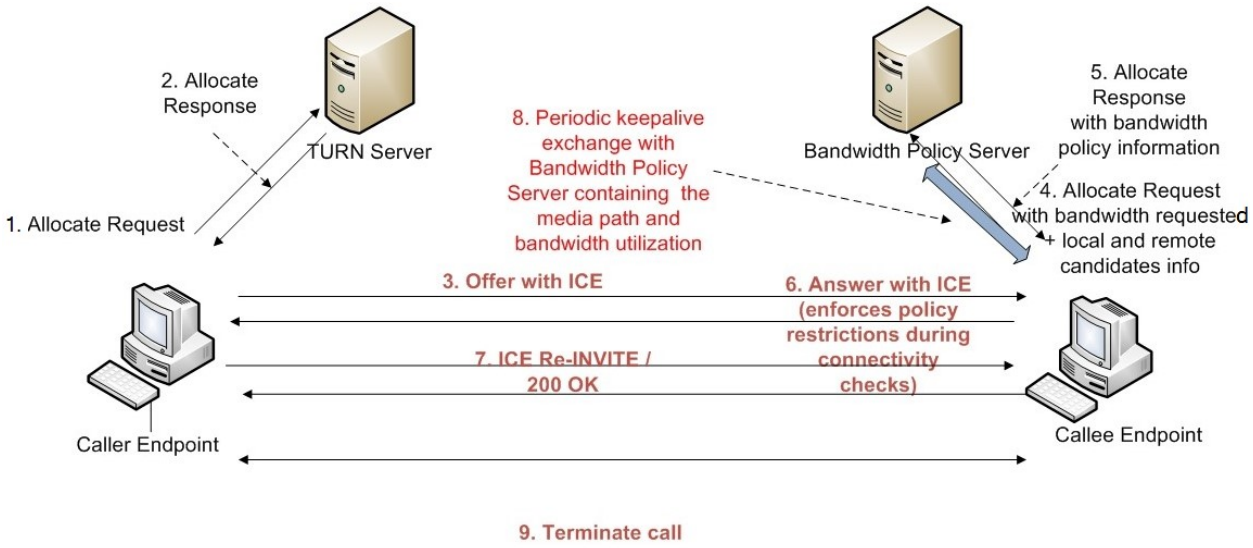


Figure 1: ICE bandwidth management

The sequence diagram in the following figure outlines the various phases involved in establishing a session between two endpoints by using both the protocol as described in [MS-ICE2] and this protocol during the different phases.

The **candidates** gathering phase is the exchange of gathered **transport addresses** between the caller and callee endpoints.

The **connectivity checks** phase is the exchange of candidates selected by the candidates gathering phase.

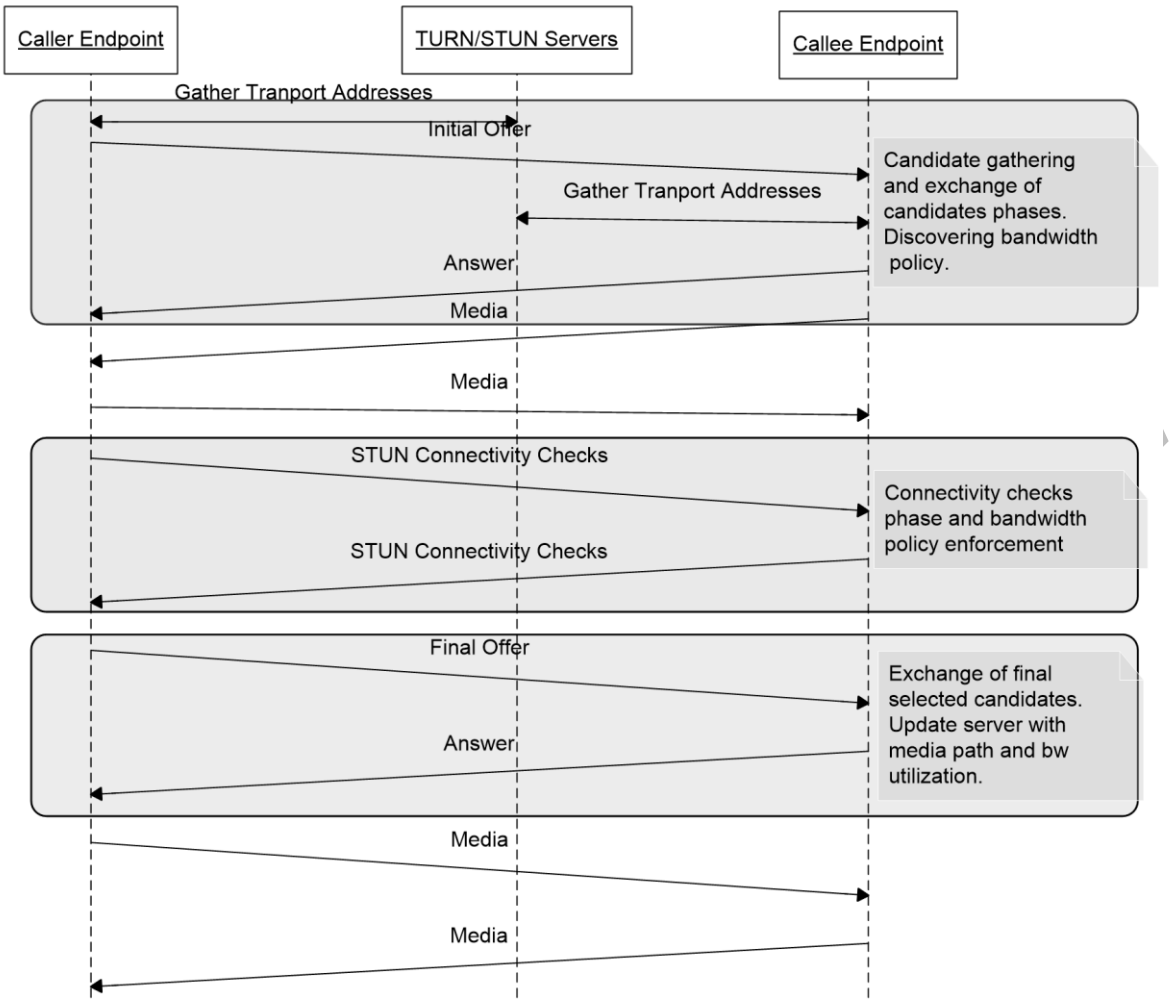


Figure 2: ICE sequence diagram

During the candidates gathering phase, the caller attempts to establish a media session and gathers transport addresses that can potentially be used to communicate with its **peer**.

The gathered transport addresses are used to form candidates. The gathered candidates are then sent to the peer in the **offer**. Typically the caller endpoint serves as the **controlling agent** or endpoint and is responsible for selecting the final candidates for media flow.

The callee endpoint is typically assigned as the **bandwidth management endpoint** by the applications. If the bandwidth management endpoint is aware of the **remote endpoint's** candidates during its candidates gathering phase, it sends the information about the bandwidth requirement for the media session along with the local **Host Candidate** and **remote candidates** to the server that it uses to gather **Relayed Candidates**. The communication between the bandwidth management endpoint and the bandwidth policy server uses the protocol described in [MS-TURNBWM]. The bandwidth policy server, if capable of bandwidth management, returns the bandwidth policy. At this point, the callee endpoint is aware of the bandwidth policy and can enforce the policy during connectivity checks. The bandwidth policy from the server returns the send and receive bandwidth for local site addresses as described in [MS-TURNBWM] section 2.2.6, local relay site addresses as

described in [MS-TURNBWM] section 2.2.7, remote site addresses as described in [MS-TURNBWM] section 2.2.4, and remote relay site addresses as described in [MS-TURNBWM] section 2.2.5. If the requested bandwidth is not available for a particular site, the candidates belonging to the site are not used for media flow. If no **candidate pairs** can be formed by the bandwidth management endpoint, it fails the media session. If the local relay site candidates cannot be used, the bandwidth management endpoint does not include the local relay site candidates in the offer or **answer**.

The gathered candidates are encoded and sent to the caller in the answer. With the exchange of candidates complete, both the endpoints are now aware of their peer's candidates. The start of the connectivity checks phase is triggered at an endpoint when it is aware of its peer's candidates. Both endpoints pair up the local and remote candidates to form a **Check List** of candidate pairs that are ordered based on the priorities of the candidate pairs. Both endpoints systematically perform connectivity checks starting from the top of the candidate pair Check List to determine the highest priority candidate pair that can be used by the endpoints to establish a media session.

Connectivity checks involve sending peer-to-peer **Simple Traversal of UDP through NAT (STUN)** binding request messages and responses from the **local transport addresses** to the remote transport addresses of each candidate pair in the list. During the connectivity checks, the bandwidth management endpoint does not attempt connectivity checks for candidate pairs that have either the local or remote candidate restricted by the bandwidth policy. In addition, if the bandwidth management endpoint receives connectivity check packets from the peer endpoint for any of the bandwidth policy restricted candidate pairs, it responds with an error response with specific error codes. The peer endpoint based on the error code disables the candidate pair, or all associated candidate pairs, for the **local candidate** for which the error response was received. This ensures that the connectivity checks do not select any of the candidates belonging to sites restricted by the bandwidth policy.

The controlling agent concludes the connectivity checks by nominating a valid candidate pair found by connectivity checks for media flow. At the end of the connectivity checks, the bandwidth management endpoint periodically updates the bandwidth policy server with the path being used for media flow and the bandwidth being utilized. Applications can alternatively also configure the bandwidth management endpoint to skip doing the bandwidth policy checks and to only report the path being used for media flow and bandwidth being utilized at the end of connectivity checks.

1.4 Relationship to Other Protocols

This protocol seamlessly integrates with and extends the Interactive Connectivity Establishment (ICE) extensions 2.0 as described in [MS-ICE2].

This protocol works with implementations of **Traversal Using Relay NAT (TURN)** protocols that adhere to the specifications in this protocol to create **TURN candidates** and **STUN candidates** for discovering bandwidth and reporting bandwidth utilization.

1.5 Prerequisites/Preconditions

This protocol requires that the **endpoints** be able to communicate through a signaling protocol, such as the **Session Initiation Protocol (SIP)**, to exchange **candidates**.

This protocol requires that the **bandwidth management endpoint** is configured with the bandwidth policy server IP address and port.

1.6 Applicability Statement

This protocol is designed to provide a mechanism for **bandwidth management endpoint** that is communicating with a bandwidth policy to discover bandwidth policy in the deployment and to enforce the bandwidth policy during connectivity checks.

This protocol requires a bandwidth policy server, as described in [MS-TURNBWM], to discover the bandwidth policy and report the bandwidth utilization.

If a bandwidth management endpoint does not have the **peer endpoints candidates** available during the candidates gathering phase, bandwidth policy will not be discovered. However, at the end of **connectivity checks** the bandwidth management endpoint will report the path being used for media flow and the bandwidth utilization to the bandwidth policy server. This protocol also works by using a **TURN server**, as described in [MS-TURN] section 3, instead of the bandwidth policy server. In this case, the functioning of this protocol is identical to the protocol described in [MS-ICE2] section 3 because there is no bandwidth policy available.

1.7 Versioning and Capability Negotiation

This protocol is implemented on top of the **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)** transport protocols for **Internet Protocol version 4 (IPv4)**/**Internet Protocol version 6 (IPv6)** as described in section 2.1. <1>

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses the **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)** transport protocols for **Internet Protocol version 4 (IPv4)**/**Internet Protocol version 6 (IPv6) endpoints**.<2> Applications implementing this protocol MUST NOT send messages that are greater than 1,500 bytes in length. They MUST be able to receive messages 1,500 bytes or less in length.

2.2 Message Syntax

This section specifies the various messages used by the implementation of this protocol. This includes both outgoing and incoming messages. The messages used by this protocol, and the protocols they belong to, are listed later in this section.

2.2.1 TURN/TURN Bandwidth Management Extension Messages

This protocol uses messages as defined in [MS-TURNBWM] to communicate with a bandwidth policy server, to discover bandwidth policy, and also to discover **Server Reflexive Candidates** and **Relayed Candidates** if the server supports it. The message syntax used by the **endpoint** is specified in [MS-TURNBWM] section 2.

2.2.2 STUN Messages

This protocol uses **Simple Traversal of UDP through NAT (STUN)** binding request and response messages for **connectivity checks** between the two **endpoints** and STUN binding error responses for enforcing bandwidth policy. The message formats MUST be as specified in [MS-ICE2] section 2.2.2.

This protocol adds two new error codes that are sent in the STUN binding error responses to enforce bandwidth policy restrictions. The codes and their reason phrases are defined as follows:

- **274 Disable Candidate:** All associated **candidate pairs** with the **local candidate** on which the error response is received are disabled as a result of bandwidth policy restrictions.
- **275 Disable Candidate Pair:** The candidate pair for which the error response is received is disabled as a result of bandwidth policy restrictions.

3 Protocol Details

3.1 Common Details

The procedures specified apply to both **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)** transport protocols unless the procedures explicitly specify a transport protocol.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol uses the same abstract data model as the one specified in [MS-ICE2] section 3.1.1.

3.1.2 Timers

This protocol adds two new timers in addition to the timers specified in [MS-ICE2] section 3.1.2.

The **ICE Bandwidth Commit** timer tracks the spacing of the bandwidth commit message sent to the bandwidth policy server. This timer MUST have a default value of 500 milliseconds.

The **ICE Bandwidth Update** timer tracks the spacing of the bandwidth update message sent to the bandwidth policy server. This timer MUST have a default value of 19 seconds or less.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

This section outlines the higher-layer events that trigger the start of the various phases of this protocol for bandwidth management. The higher-layer triggered events as specified in [MS-ICE2] section 3.1.4 MUST be followed in addition to the specifications in the following sections. Differences in processing from the specifications as specified in [MS-ICE2] section 3.1.4 are specified in the corresponding sections. Discovering the bandwidth policy and enforcing and reporting bandwidth utilization MUST be carried out independently for each media stream.

The **callee endpoint** is typically configured as the **bandwidth management endpoint** because the callee endpoint, on receiving the **initial offer**, knows the **candidates** of the **peer** endpoint. However, in cases where the **caller** is an Interactive Connectivity Establishment (ICE) endpoint and the peer endpoint is a non-ICE endpoint, applications can configure the caller endpoint as the bandwidth management endpoint. The bandwidth management endpoint is responsible for bandwidth policy discovery, policy enforcement, and reporting for that media session. Applications can configure the bandwidth management endpoint to skip the bandwidth policy discovery and policy enforcement and to only do the reporting for that media session.

3.1.4.1 Sending the Initial Offer

The **caller** attempting to establish a media session with a **peer** MUST follow the procedures as specified in [MS-ICE2] section 3.1.4.1.

In addition, if the caller **endpoint** is a **bandwidth management endpoint** and is aware of the peer endpoints site addresses, the caller endpoint MUST perform the checks for bandwidth admission control during its **candidates** gathering phase, as specified in [MS-TURNBWM] section 3.2.4.1, to determine the bandwidth policy. If the bandwidth policy from the bandwidth policy server restricts all potential paths between the caller and the **callee** endpoint, the attempt to establish the media session MUST fail and the **offer** MUST NOT be sent.

If the policy received from the bandwidth policy server restricts the use of the local **Relayed Candidate**, the local Relayed Candidate MUST NOT be included in the offer.

3.1.4.2 Receiving the Initial Offer and Generating the Answer

The **callee endpoint**, on receiving the **initial offer**, MUST follow the procedures specified in [MS-ICE2] section 3.1.4.2.

If the callee endpoint is a **bandwidth management endpoint**, and has been configured to do bandwidth policy discovery and policy enforcement, the callee endpoint MUST perform bandwidth policy checks during its **candidates** gathering phase, as specified in section 3.1.4.8.1.1, by using the **remote endpoints** candidate information received in the **offer**. If the bandwidth policy from the bandwidth policy server restricts all potential paths between the **caller** and the callee endpoint, the attempt to establish the media session MUST fail. If the policy received from the bandwidth policy server restricts the use of the local **Relayed Candidate**, the local Relayed Candidate MUST NOT be included in the **answer**.

3.1.4.3 Processing the Provisional Answer to the Initial Offer

The **caller**, after receiving the **provisional answer** with the **callee's candidates** MUST follow the procedures as specified in [MS-ICE2] section 3.1.4.3.

3.1.4.4 Processing the Answer to the Initial Offer

Answer processing depends on the type of **peer**.

3.1.4.4.1 Processing the Answer to the Initial Offer from a Full ICE Peer

If an **answer** is received from a **full ICE peer**, the procedure as specified in [MS-ICE2] section 3.1.4.4 MUST be followed.

3.1.4.4.2 Processing the Answer to the Initial Offer from a Non-ICE or Lite Peer

If an **answer** is received from a non-ICE or **Lite peer**, the procedure as specified in [MS-ICE2] section 3.1.4.4.1 MUST be followed.

If the **endpoint** is a **bandwidth management endpoint**, the endpoint MUST follow the procedures specified in section 3.1.4.8.2.2 and section 3.1.4.8.2.3 for committing and sending periodic updates to the bandwidth policy server.

3.1.4.5 Generating the Final Offer

At the end of the **connectivity checks** phase, the controlling **endpoint** MUST follow the procedures as specified in [MS-ICE2] section 3.1.4.5.

If the endpoint is a **bandwidth management endpoint**, it MUST use the procedures specified in section 3.1.4.8.2.2 for committing, and in section 3.1.4.8.2.3 for sending, periodic updates to the bandwidth policy server.

3.1.4.6 Receiving the Final Offer and Generating the Answer

The **controlled agent**, on receiving the **final offer**, MUST follow the procedures as specified in [MS-ICE2] section 3.1.4.6.

If the **endpoint** is a **bandwidth management endpoint**, it MUST use the procedures specified in section 3.1.4.8.2.2 for committing, and in section 3.1.4.8.2.3 for sending, periodic updates to the bandwidth policy server.

3.1.4.7 Processing the Answer to the Final Offer

The **controlling agent**, after receiving the **answer** to its **final offer**, MUST follow the procedures as specified in [MS-ICE2] section 3.1.4.7.

If the **endpoint** is a **bandwidth management endpoint**, it MUST follow the procedures specified in section 3.1.4.8.2.2 and section 3.1.4.8.2.3 to commit and send periodic updates to the bandwidth policy server.

3.1.4.8 Common Procedures

Bandwidth management endpoints, when configured with a bandwidth policy server, follow the specifications as specified in [MS-TURNBWM] section 2.

3.1.4.8.1 Candidates Gathering Phase

The **candidates** gathering phase MUST follow the implementation requirements specified in [MS-ICE2] section 3.1.4.8.1. If the **endpoint** is a **bandwidth management endpoint** and is aware of the site addresses of the **peer's** endpoints, it MUST follow the procedures specified in section 3.1.4.8.1.1 for the bandwidth policy checks.

3.1.4.8.1.1 Bandwidth Admission Check Request

The **bandwidth management endpoint** SHOULD perform the policy checks by following the procedures as specified in [MS-TURNBWM] section 3.2.4.1.

The **Local Site Address** attribute MUST be populated with one of the local **Host Candidate** addresses. The **Remote Site Address** attribute MUST be populated with any one Host Candidate addresses of the **peer endpoint**. If the peer endpoint has a **Relayed Candidate**, the **Remote Relay Site Address** attribute MUST be populated with the address of the Relayed Candidate. If both IPv4 and IPv6 Host Candidate or Relayed Candidate addresses are available then IPv4 addresses MUST be used to populate the respective **Site Address** attributes for the bandwidth checks.

If the endpoint is configured with both **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)** bandwidth policy servers, the checks MUST be sent to both servers. The first valid check response from the server MUST be honored and subsequent check responses from the servers MUST be ignored.

3.1.4.8.2 Connectivity Checks Phase

The procedures as specified in [MS-ICE2] section 3.1.4.8.2 for the **connectivity checks** MUST be followed. This section specifies additional functionality for bandwidth management. At the end of the connectivity checks phase, the **bandwidth management endpoint** MUST commit the **candidates** selected by the connectivity checks, and the bandwidth used, to the policy server. For the duration of the media session, the bandwidth management endpoint MUST send periodic updates to the policy server based on the **ICE Bandwidth Update** timer. If both **User Datagram Protocol (UDP)** and **Transmission Control Protocol (TCP)** bandwidth policy servers are available, the UDP bandwidth policy server SHOULD be used for commits and updates unless media is flowing over a TCP **Relayed**

Candidate for the bandwidth management endpoint. In that case, the TCP bandwidth policy server SHOULD be used for commits and updates.

3.1.4.8.2.1 Formation of Candidate Pairs

The **candidate pairs** MUST be formed as specified in [MS-ICE2] section 3.1.4.8.2.1. If the **bandwidth management endpoint** has received a policy response from the policy server, candidate pairs that have either the **local candidate** or the **remote candidate** invalidated by the bandwidth policy MUST be considered invalidated and **connectivity check** requests MUST NOT be sent for those candidate pairs.

3.1.4.8.2.2 Bandwidth Admission Commit Request

At the end of successful **connectivity checks**, the **bandwidth management endpoint** sends the commit request by following the procedures as specified in [MS-TURNBWM] section 3.2.4.2 when the **ICE Bandwidth Commit** timer fires.

If the **local candidate** selected by the connectivity checks is not a **Relayed Candidate**, the **Local Site Address** attribute MUST be populated with the **candidate** address of a local candidate selected by the connectivity checks.

If the local candidate selected by the connectivity checks is a Relayed Candidate, the **Local Site Address** attribute MUST be populated with the candidate address of the local **Host Candidate** that was used to gather the Relayed Candidate. The **Local Relay Site Address** attribute, as specified in [MS-TURNBWM] section 2.2.7, MUST be populated with the relay address of the selected local candidate.

If the **remote candidate** selected by the connectivity checks is not a Relayed Candidate, the **Remote Site Address** attribute MUST be populated with the candidate address of the remote candidate selected by the connectivity checks.

If the remote candidate selected by the connectivity checks is a Relayed Candidate, the **Remote Site Address** attribute MUST be populated with the candidate address of any remote Host Candidate. If both IPv4 and IPv6 remote Host Candidate addresses are available then an IPv4 Host Candidate address MUST be used to populate the **Remote Site Address** attribute. The **Remote Relay Site Address** attribute MUST be populated with the relay address of the selected remote candidate.

3.1.4.8.2.3 Bandwidth Admission Update Request

After receiving a valid commit response, the **bandwidth management endpoint** MUST send the update request by following the procedures as specified in [MS-TURNBWM] section 3.2.4.3 when the **ICE Bandwidth Update** timer fires.

The bandwidth management endpoint SHOULD generate an update request if the bandwidth value is different from the bandwidth value that was sent in the commit request, by using the update request procedures as specified in [MS-TURNBWM] section 3.2.4.3 when the **ICE Bandwidth Update** timer fires.

3.1.4.8.3 Media Flow

The **candidate pairs** to be used for media flow MUST follow the specifications as specified in [MS-ICE2] section 3.1.4.8.3.

3.1.5 Message Processing Events and Sequencing Rules

The processing of messages and sequencing rules MUST be as specified in [MS-ICE2] section 3.1.5. This section specifies additional procedures for the **endpoints** for bandwidth management.

3.1.5.1 Processing TURN Bandwidth Management Extensions Messages

This section specifies the message processing and sequencing rules for the **Traversal Using Relay NAT (TURN)** bandwidth management extensions.

3.1.5.1.1 Processing a Bandwidth Check Response

The format of the check response from the server and its attributes are specified in [MS-TURNBWM] section 2. If a site address response does not have the **valid** flag set, the **candidates** belonging to the site MUST be considered restricted. If the **valid** flag is set, the site address response also provides the send and receive bandwidths that can be used by the candidates belonging to a particular site when the application uses those candidates for media flow.

The local site address response applies to local **Host Candidates**, **Server Reflexive Candidates**, and their associated **peer-derived candidates**. The local relay site address response applies to the local **Relayed Candidates** and their associated Server Reflexive Candidates.

The remote site address response decision applies to remote Host Candidates, remote Server Reflexive Candidates, and their associated peer-derived candidates. The remote relay site address response applies to the remote Relayed Candidates and their associated Server Reflexive Candidates.

3.1.5.1.2 Processing a Bandwidth Commit Response

The format of the commit response from the server and its attributes are specified in [MS-TURNBWM] section 2. The reservation identifier provided by the bandwidth policy server MUST be included by the **endpoint** in subsequent update messages sent to the policy server.

If the server returns a NULL reservation identifier, further bandwidth messages MUST NOT be sent to the bandwidth policy server.

3.1.5.1.3 Processing a Bandwidth Update Response

The format of the update response from the server and its processing MUST be as specified in [MS-TURNBWM] section 2 and 3.2.5.3.

3.1.5.2 Processing STUN Connectivity Check Messages

The processing of **Simple Traversal of UDP through NAT (STUN) connectivity checks** MUST be as specified in [MS-ICE2] section 3.1.5.2. This section specifies additional procedures for the **endpoints** for bandwidth management.

3.1.5.2.1 Processing a STUN Binding Request

If the **bandwidth management endpoint** receives a **Simple Traversal of UDP through NAT (STUN)** binding request from a **remote candidate** that has been restricted by the bandwidth policy, the **endpoint** MUST send a STUN binding error response with the error code set to 274 Disable Candidate.

If a STUN binding request is received for a **local candidate** that is restricted by the bandwidth policy and the remote candidate is not restricted by bandwidth policy, the endpoint MUST send a STUN binding error response with the error code set to 275 Disable Candidate Pair. The remote candidate type can be determined based on the **Candidate Identifier** attribute present in the STUN binding request messages.

3.1.5.2.2 Processing a STUN Binding Error Response

If a valid error response is received with a 274 Disable Candidate error code, the **endpoint** MUST set all **candidate pairs** associated with the **local candidate** on which the error response is received to the "Failed" state.

If a valid error response is received with a 275 Disable Candidate Pair error code, the endpoint MUST set the candidate pairs for which the error response is received to the "Failed" state.

3.1.6 Timer Events

All the timer events as specified in [MS-ICE2] section 3.1.6 MUST be followed in addition to the timer events specified in this section.

3.1.6.1 ICE Bandwidth Commit Timer

The **ICE Bandwidth Commit** timer MUST fire every 500 milliseconds on the **bandwidth management endpoint** until a bandwidth commit response is received from the server. When this timer fires, the bandwidth commit request MUST be sent as specified in section 3.1.4.8.2.2.

3.1.6.2 ICE Bandwidth Update Timer

The **ICE Bandwidth Update** timer MUST fire on the **bandwidth management endpoint** every 19 seconds or less after the bandwidth has been committed to the policy server. When the timer fires, a bandwidth update message MUST be sent, as specified in section 3.1.4.8.2.3, if the bandwidth being updated is same as the bandwidth that was sent in the commit request. This facilitates the bandwidth server to track the bandwidth utilization for the media session.

If the update request is for a bandwidth value that is different from the bandwidth sent in the commit, this timer MUST fire every second until an update response is received from the server. After the update response is received from the server, the timer MUST switch to firing every 19 seconds or less.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following figure shows a sample deployment scenario for bandwidth management.

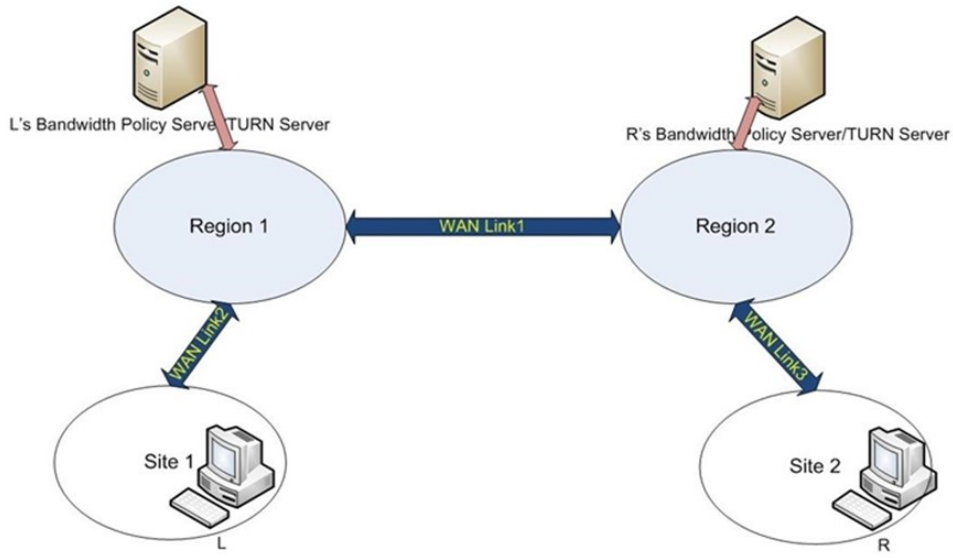


Figure 3: Bandwidth management deployment scenario

In the preceding figure, Endpoint L and Endpoint R belong to different sites that are connected via wide area network (WAN) links that are under bandwidth management. Both Endpoint L and Endpoint R are configured with their respective **User Datagram Protocol (UDP)** bandwidth policy servers that provide **Traversal Using Relay NAT (TURN)** functionality, in addition to bandwidth policy management servers, LPS for Endpoint L and RPS for Endpoint R.

Both **agents** are **full** Interactive Connectivity Establishment (ICE) implementations and use regular nominations for selecting the **candidates** to be used for media flow. In the following example, WAN Link 1 is congested and does not have bandwidth available for media flow. WAN Link 2 and WAN Link 3 have bandwidth available for media flow.

The **transport address** follows a similar naming convention to the sample as described in [MS-ICE2] section 4.

Transport addresses are referred to by using mnemonic names with the format *entity-type-seqno*, where *entity* refers to the entity whose IP address the transport address is on, and is one of "L", "R", "LPS", or "RPS". The *type* is either "PUB" for transport addresses that are publicly reachable on the Internet or "PRIV" for transport addresses that are not reachable from the Internet. The *seqno* is a number that is different for transport addresses of the same type on an entity.

Endpoint L has a private address L-PRIV-1 (192.168.2.1) and Endpoint R has a private address R-PRIV-1 (192.157.2.1).

LPS has a transport address LPS-PRIV-1 on the private edge (192.170.20.1) and external edge with LPS-PUB-1 (10.101.0.57).

RPS has a transport address RPS-PRIV-1 on the internal edge (192.175.54.2) and external edge with RPS-PUB-1 (10.107.0.37).

Definitions for the call flow are as follows:

- "S=" refers to the source transport address.
- "D=" refers to the destination transport address.
- "SD=" refers to the destination address to which the **TURN server** has to forward the packet.
- "LSA=" refers to the **local site address** attribute.
- "LRA=" refers to the **local relay site address** attribute.
- "RSA=" refers to the **remote site address** attribute.
- "RRA=" refers to the **remote relay site address** attribute.
- "USE-CAND" implies the presence of the **USE-CANDIDATE** attribute, as described in [IETFDRAFT-ICENAT-19] section 7.1.1.1.
- "DIS-CAND" implies the presence of the Disable Candidate error code in the message.
- "DIS-LS" implies that the bandwidth policy server disallows the usage of candidates belonging to the local site.
- "DIS-RS" implies that the bandwidth policy server disallows the usage of candidates belonging to the remote site.
- "BW-CHK-REQ" implies the presence of bandwidth admission check request attributes.
- "BW-CHK-RES" implies the presence of bandwidth admission check response attributes.
- "BW-CMT-REQ" implies the presence of bandwidth admission commit request attributes.
- "BW-CMT-RES" implies the presence of bandwidth admission commit response attributes.
- "RES-ID" implies the presence of the **reservation ID** attribute.
- "BW-UPD-REQ" implies the presence of bandwidth admission update request attributes.
- "MA=" refers to the mapped address in the **Simple Traversal of UDP through NAT (STUN)** binding response.
- "RA=" refers to the reflexive address.
- "TA=" refers to the relay transport address.

For clarity, the example does not show the TURN authentication mechanisms and the **Real-Time Transport Control Protocol (RTCP) component**.

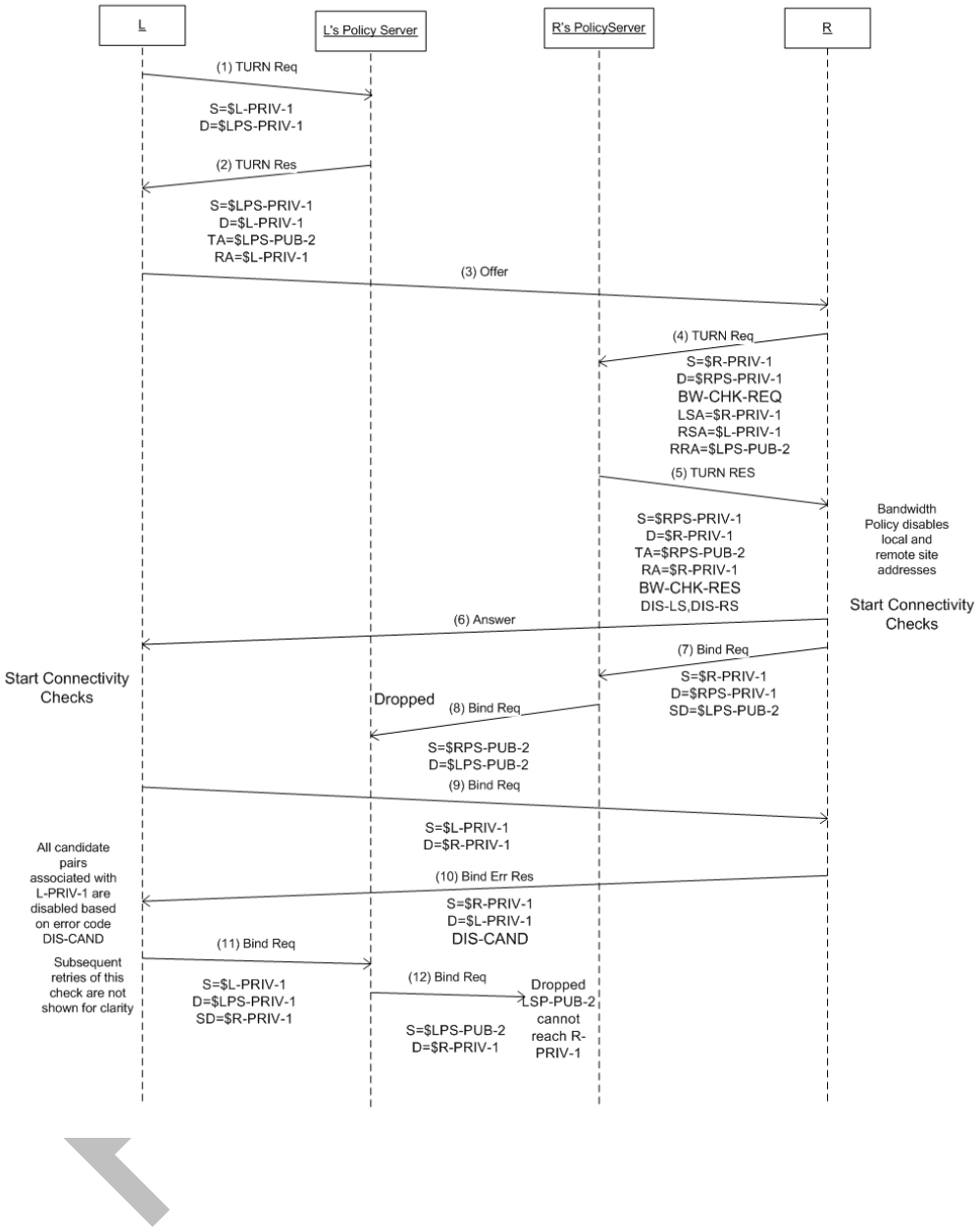
The example focuses on the **Real-Time Transport Protocol (RTP)** component for establishing a media session between Endpoint L and Endpoint R with bandwidth policy management and does not focus on protocol details as described in [MS-TURNBWM].

Endpoint L initiates the media session and becomes the **controlling agent** because Endpoint L is a full ICE implementation. Endpoint L gathers its UDP **Host Candidate** by binding to its local interface and then gathers a UDP **Relayed Candidate** from the configured server, LPS. Because no **Transmission Control Protocol (TCP)** TURN servers are configured, Endpoint L creates a TCP-ACT **Server Reflexive Candidate** based on the UDP Host Candidate. After gathering the candidates,

Endpoint L sends the **INVITE** to Endpoint R. A sample INVITE **Session Description Protocol (SDP)** for Endpoint L's topology is as follows:

```
v=0
o=- 0 0 IN IP4 10.101.0.57
s=session
c=IN IP4 10.101.0.57
b=CT:99980
t=0 0
m=audio 52732 RTP/AVP 114 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag:qkEP
a=ice-pwd:ed6f9GuHjLcoCN6sC/Eh7fVl
a=candidate:1 1 UDP 2130706431 192.168.2.1 50005 typ host
a=candidate:2 1 UDP 16648703 10.101.0.57 52732 typ relay raddr 192.168.2.1 rport 50033
a=candidate:4 1 TCP-ACT 1684797951 192.168.2.1 50005 typ srflx raddr 192.168.2.1 rport 50005
a=rtpmap:114 x-msrta/16000
```

The following figure is the call flow for the RTP component for establishing a media session between Endpoint L and Endpoint R with bandwidth policy management.



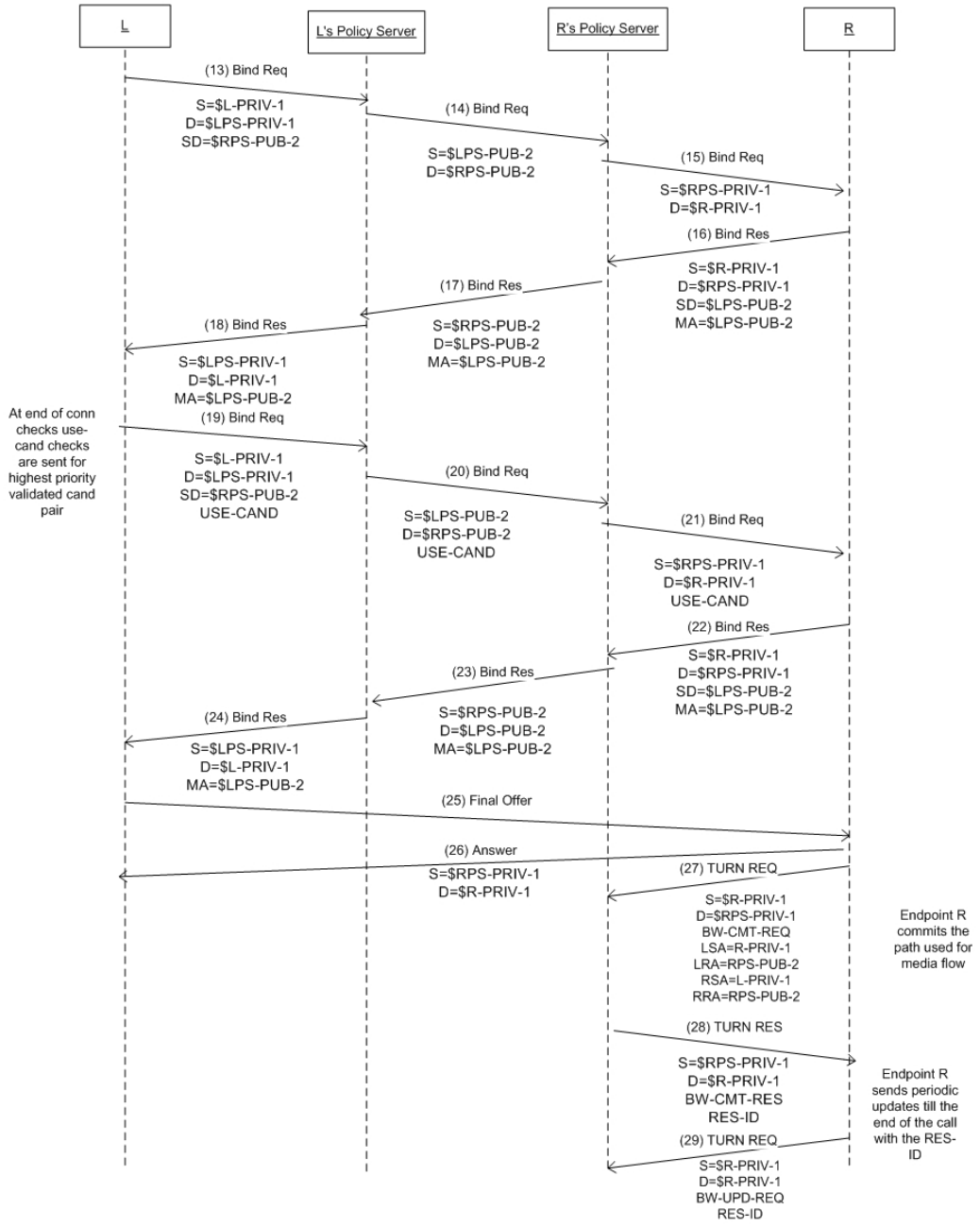


Figure 4: RTP component call flow for bandwidth management

Endpoint R, upon receiving the **offer**, gathers its candidates. Endpoint R is assigned as the **bandwidth management endpoint** for this session. At this point, Endpoint R is aware of the candidates of the **peer endpoint**. It gathers its UDP Host Candidate by binding to its local interface and then gathers the UDP Relayed Candidate from the configured bandwidth policy server. In the

allocate request sent to the bandwidth policy server endpoint at RPS-PRIV-1, Endpoint R adds bandwidth management check attributes to perform policy checks, as specified in section 3.1.4.8.1.1. Endpoint R populates the **Local Site Address** attribute with "R-PRIV-1", the **Remote Site Address** attribute with "L-PRIV-1" because Endpoint R is not behind a **network address translation (NAT)**, and the **Remote Relay Site** attribute with "LPS-PUB-1". Endpoint R specifies the bandwidth needed for this call in the **Bandwidth Reservation Amount** attribute, as described in [MS-TURNBWM] section 2.2.3. The bandwidth policy server disables both the local site and remote site address in the allocate response, which includes the **Bandwidth Check Response** attributes, because Wan Link1 does not have available bandwidth for the media session. Endpoint R gathers its Relayed Candidate "RPS-PUB-2" from the allocate response. Because no TURN TCP servers are configured, Endpoint R creates a TCP-ACT Server Reflexive Candidate based on the UDP Host Candidate.

Endpoint R, based on the policy decision received, does not form **candidate pairs** for candidates that have been disabled by the bandwidth policy. As a result of the bandwidth policy, Endpoint R has only one candidate pair, which is RPS-PUB-2 to LPS-PUB-2. A sample **answer** SDP for Endpoint R's topology is as follows:

```
v=0
o=- 0 0 IN IP4 10.107.0.37
s=session
c=IN IP4 10.107.0.37
b=CT:99980
t=0 0
m=audio 52714 RTP/AVP 114 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag:qkEP
a=ice-pwd:ed6f9GuHjLcoCN6sC/Eh7fVl
a=candidate:1 1 UDP 2130706431 192.175.54.2 50025 typ host
a=candidate:2 1 UDP 16648703 10.107.0.37 52714 typ relay raddr 192.175.54.2 rport 50036
a=candidate:3 1 TCP-ACT 1684797951 192.175.54.2 50025 typ srflx raddr 192.175.54.2 rport
50025 a=rtpmap:114 x-msrta/16000
```

Endpoint R starts **connectivity checks** for its only candidate pair and sends a STUN binding request to LPS-PUB-2 from its Relayed Candidate RPS-PUB-2, which gets dropped at LPS-PUB-2 because permissions have not been opened for RPS-PUB-2 at LPS-PUB-2. This STUN binding request results in permission being opened for LPS-PUB-2 at RPS-PUB-2.

Endpoint L, on receiving the answer, pairs up its candidates with Endpoint R's candidates received in the answer and starts connectivity checks with the highest priority candidate pair. Endpoint L sends a STUN binding request from L-PRIV-1 to R-PRIV-1. Endpoint R, on receiving this STUN binding request from L-PRIV-1, sends a STUN binding error response with the Disable Candidate error code because both the local site address and the remote site address have been disabled as a result of bandwidth policy and cannot be used for media flow. Endpoint L, on receiving the STUN binding error response, disables all candidate pairs whose **local candidates** belong to the local site, including Host Candidates, Server Reflexive Candidates, or local **peer-derived candidates**.

Endpoint L sends a STUN binding request from its Relayed Candidate LPS-PUB-2 to R-PRIV-1, which gets dropped because R-PRIV-1 is not reachable from the public interface. Endpoint L then sends a STUN binding request from its Relayed Candidate LPS-PUB-2 to RPS-PUB-2, which Endpoint R receives from its Relayed Candidate because permissions have already been opened on RPS-PUB-2 for LPS-PUB-2. Endpoint L, on receiving the STUN binding response, validates this candidate pair. At the end of the connectivity checks timeout, Endpoint L nominates its only valid candidate pair and sends a STUN binding request with the **USE-CANDIDATE** attribute (as described in [IETFDRAFT-ICENAT-19] section 7.1.1.1) set. On getting the response, Endpoint L sends the **final offer** to the endpoint with the final candidates to be used for media flow. A sample SDP for the final offer is as follows:

```
v=0
o=- 0 0 IN IP4 10.101.0.57
s=session
c=IN IP4 10.101.0.57
b=CT:99980
```

```
t=0 0
m=audio 52732 RTP/SAVP 114 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag:32sD
a=ice-pwd:YF9/OwRcN/pXUglBv1c+5QMu
a=candidate:1  UDP 16648703 10.101.0.57 52732 typ relay raddr 192.168.2.1 rport 50033
a=remote-candidates:1 10.107.0.37 52714
a=rtpmap:114 x-msrta/16000
```

Endpoint R, on receiving the final offer, sends the answer to the final offer. A sample SDP for the final offer is as follows:

```
v=0
o=- 0 0 IN IP4 10.107.0.37
s=session
c=IN IP4 10.107.0.37
b=CT:99980
t=0 0
m=audio 52714 5 RTP/SAVP 114 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag:32sD
a=ice-pwd:YF9/OwRcN/pXUglBv1c+5QMu
a=candidate:1  UDP 16648703 10.107.0.37 52714 typ relay raddr 192.175.54.2 rport 50036
a=remote-candidates:1 101.0.57 52732
a=rtpmap:114 x-msrta/16000
```

Endpoint R is the bandwidth management endpoint that also sends a Bandwidth Commit message to the relay to notify the policy server that the candidates are being used for media flow. Endpoint R populates LSA with "R-PRIV-1", LRA with "RPS-PUB-2", RSA with "L-PRIV-1", and RRA with "LPS-PUB-2" because both endpoints are using their Relayed Candidates for media flow.

Endpoint R, on receiving the bandwidth admission commit response with a **reservation ID** from RPS, starts to send periodic bandwidth admission update requests to RPS for the duration of the media session with the **reservation ID** received in the commit response added to every bandwidth admission update request.

5 Security

5.1 Security Considerations for Implementers

This protocol has similar security concerns as those described in [MS-ICE2] section 5. Additional considerations and mitigations for this protocol are listed in this section.

5.1.1 Attacks on Bandwidth Policy Processing

The security considerations for determining the bandwidth policy, as described in [MS-TURNBWM], are described in [MS-TURNBWM] section 5.1.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Lync 2010
- Microsoft Lync Server 2010
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Lync Server 2013
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015
- Microsoft Skype for Business 2019 Preview
- Microsoft Skype for Business Server 2019 Preview

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 1.7: Lync 2010, Lync Server 2010: IPV6 is not supported.

<2> Section 2.1: Lync 2010, Lync Server 2010: IPV6 is not supported.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
6 Appendix A: Product Behavior	Updated list of supported products.	major

8 Index

V