Network Working Group
Request for Comments: 4961
BCP: 131
Category: Best Current Practice

D. Wing
Cisco Systems

July 2007

Symmetric RTP / RTP Control Protocol (RTCP)

Status of This Memo

Copyright Notice

Abstract

This document recommends using one UDP port pair for both communication directions of bidirectional RTP and RTP Control Protocol (RTCP) sessions, commonly called "symmetric RTP" and "symmetric RTCP".

Table of Contents

1. Introduction 2
2. Conventions Used in this Document 2
3. Definition of Symmetric RTP and Symmetric RTCP 3
4. Recommended Usage 3
5. Security Considerations 4
6. Acknowledgments 4
7. References 4
   7.1. Normative References 4
   7.2. Informative References 4

1. Introduction

TCP [RFC0793], which is inherently bidirectional, transmits and receives data using the same local port. That is, when a TCP connection is established from host A with source TCP port "a" to a remote host, the remote host sends packets back to host A's source TCP port "a".

However, UDP is not inherently bidirectional and UDP does not require using the same port for sending and receiving bidirectional traffic. Rather, some UDP applications use a single UDP port to transmit and receive (e.g., DNS [RFC1035]), some applications use different UDP ports to transmit and receive with explicit signaling (e.g., Trivial File Transfer Protocol (TFTP) [RFC1350]), and other applications don't specify the choice of transmit and receive ports (RTP [RFC3550]).

Because RTP and RTCP are not inherently bidirectional protocols, and UDP is not a bidirectional protocol, the usefulness of using the same UDP port for transmitting and receiving has been generally ignored for RTP and RTCP. Many firewalls, Network Address Translators (NATs) [RFC3022], and RTP implementations expect symmetric RTP, and do not work in the presence of asymmetric RTP. However, this term has never been defined. This document defines "symmetric RTP" and "symmetric RTCP".

The UDP port number to receive media, and the UDP port to transmit media are both selected by the device that receives that media and transmits that media. For unicast flows, the receive port is communicated to the remote peer (e.g., Session Description Protocol (SDP) [RFC4566] carried in SIP [RFC3261], Session Announcement Protocol (SAP) [RFC2974], or Megaco/H.248 [RFC3525]).

There is no correspondence between the local RTP (or RTCP) port and the remote RTP (or RTCP) port. That is, device "A" might choose its local transmit and receive port to be 1234. Its peer, device "B", is not constrained to also use port 1234 for its port. In fact, such a constraint is impossible to meet because device "B" might already be using that port for another application.

The benefits of using one UDP port pair is described below in Section 4.

2. Conventions Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Definition of Symmetric RTP and Symmetric RTCP

A device supports symmetric RTP if it selects, communicates, and uses IP addresses and port numbers such that, when receiving a bidirectional RTP media stream on UDP port "A" and IP address "a", it also transmits RTP media for that stream from the same source UDP port "A" and IP address "a". That is, it uses the same UDP port to transmit and receive one RTP stream.

A device that doesn't support symmetric RTP would transmit RTP from a different port, or from a different IP address, than the port and IP address used to receive RTP for that bidirectional media steam.

A device supports symmetric RTCP if it selects, communicates, and uses IP addresses and port numbers such that, when receiving RTCP packets for a media stream on UDP port "B" and IP address "b", it also transmits RTCP packets for that stream from the same source UDP port "B" and IP address "b". That is, it uses the same UDP port to transmit and receive one RTCP stream.

A device that doesn't support symmetric RTCP would transmit RTCP from a different port, or from a different IP address, than the port and IP address used to receive RTCP.

4. Recommended Usage

There are two specific instances where symmetric RTP and symmetric RTCP are REQUIRED:

The first instance is NATs that lack integrated Application Layer Gateway (ALG) functionality. Such NATs require that endpoints use symmetric UDP ports to establish bidirectional traffic. This requirement exists for all types of NATs described in Section 4 of [RFC4787]. ALGs are defined in Section 4.4 of [RFC3022].

The second instance is Session Border Controllers (SBCs) and other forms of RTP and RTCP relays (e.g., [TURN]). Media relays are necessary to establish bidirectional UDP communication across a NAT that is 'Address-Dependent' or 'Address and Port-Dependent' [RFC4787]. However, even with a media relay, symmetric UDP ports are still required to traverse such a NAT.

There are other instances where symmetric RTP and symmetric RTCP are helpful, but not required. For example, if a firewall can expect symmetric RTP and symmetric RTCP, then the firewall's dynamic per-call port filter list can be more restrictive compared to asymmetric RTP and asymmetric RTCP. Symmetric RTP and symmetric RTCP can also ease debugging and troubleshooting.

Other UDP-based protocols can also benefit from common local transmit and receive ports.

There are no known cases where symmetric RTP or symmetric RTCP are harmful.

For these reasons, it is RECOMMENDED that symmetric RTP and symmetric RTCP always be used for bidirectional RTP media streams.

5. Security Considerations

If an attacker learns the source and destination UDP ports of a symmetric RTP or symmetric RTCP flow, the attacker can send RTP or RTCP packets to that host. This differs from asymmetric RTP and asymmetric RTCP, where an attacker has to learn the UDP source and destination ports used for the reverse traffic, before it can send packets to that host. Thus, if a host uses symmetric RTP or symmetric RTCP, an attacker need only see one RTP or RTCP packet in order to attack either RTP endpoint. Note that this attack is similar to that of other UDP-based protocols that use one UDP port pair (e.g., DNS [RFC1035]).

6. Acknowledgments

The author thanks Francois Audet, Sunil Bhargo, Lars Eggert, Francois Le Faucheur, Cullen Jennings, Benny Rodrig, Robert Sparks, and Joe Stone for their assistance with this document.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, July 1992.

[TURN] Rosenberg, J., "Obtaining Relay Addresses from Simple Traversal Underneath NAT (STUN)", Work in Progress, July 2007.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[RFC3525] Groves, C., Pantaleo, M., Anderson, T., and T. Taylor, "Gateway Control Protocol Version 1", RFC 3525, June 2003.

Author’s Address

Dan Wing
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

EMail: dwing@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.