H.323 FORUM

Open Community Specification

File Transfer in H.323 Systems

4 October 2009

IMTC

International Multimedia Communications Consortium

Summary

A trivial file transfer capability for the transmitting of stored textual, multimedia or binary information over an RTP connection. This specification maybe referred to as TFTP over RTP.

Author(s)

Simon Horne, Spranto Australia Pty. Ltd.

Acknowledgements

Table of Contents

File Transfer in H.323 Systems

1 Scope

This recommendation defines the capabilities and procedures for facilitating file transfer between H.323 Endpoints. The File Transfer Capability uses TFTP (RFC 1350) to transfer stored textual, multimedia or binary information within the context of an RTP session.

2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation H.245 (2006), Control Protocol for Multimedia Communications.

[2] IETF RFC 1350 (1992), The TFTP Protocol (Revision 2).

[3] IETF RFC 2348 (1998), TFTP Timeout Interval and Transfer Size Options.

3 Capability Description

This specification described a method to provide for the facilitation of transmission or receiving of stored textual, multimedia or binary information via the use of Trivial File Transfer Protocol (v2) or TFTP (RFC 1350) to deliver content over an RTP session within a H.323 system for the purpose of storage or playback on a remote device.

4 Trivial File Transfer Capability Definitions

The capability identifier defined in Table 1 is allocated to provide support for the TFTP Capability as defined in RFC 1350. Table 2 define the associated capability block sizes as allowed in the associated RFC 2348. Table 3 defines the mode of data transfer.

For the purpose of this capability the TFTP mode shall be “octet” to specify binary file transfer and shall also support RFC 2348 and RFC 2349.

Table 1 – Capability Identifier for TFTP Capability

Capability name	TFTP
Capability class	Generic Data Capability
Capability identifier type	OID

Capability identifier value	iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) packetizer(17090) h245 (1) tftp_rtp (2)
maxBitRate	Mandatory.
collapsing	This field may be present and consisting of the parameters defined below.
nonCollapsing	This field shall not be included and shall be ignored if received.
nonCollapsingRaw	This field shall not be included
transport	This field shall not be included

Table 2 – TFTP Capability Parameter – Block Size

Parameter name	BlockSize
Parameter description	This is a collapsing booleanArray GenericParameter which specifies the tftp block sizes available (in accordance to RFC2348). Each bit represents a block size and Bits 1 – 8 will be set depending on the block size supported. Bit 8 (value 1) – If set, indicates 512 octets Bit 7 (value 2) – If set, indicates 1024 octets Bit 6 (value 4) – It set, indicates 1428 octets Bit 5 (value 8) – If set, indicates 2048 octets Bit 4 (value 16) – If set, indicates 4096 octets Bit 3 (value 32) – If set, indicates 8192 octets Bit 2 (value 64) – If set, indicates 16384 octets Bit 1 (value 128) – If set, indicates 32768 octets In capability exchange, this parameter indicates supported block size and in logical channel signaling, indicates the block size to be used for the current session.
Parameter identifier value	1
Parameter status	Mandatory
Parameter type	booleanArray
Supersedes	–

Table 3 – TFTP Capability Parameter – Transfer Mode

Parameter name	Transfer Mode
Parameter description	This is a collapsing booleanArray GenericParameter which specifies the Transfer Mode available. Each bit represents a transfer mode and Bits 1 – 8 will be set depending on the transfer mode supported. Bit 8 (value 1) – If set, indicates RTP encapsulation Bit 7 (value 2) – If set, indicates Raw – No encapsulation
Parameter identifier value	2
Parameter status	Optional - If absent, it is assumed that the Transfer mode is RTP encapsulated.
Parameter type	booleanArray
Supersedes	–

4.1 TFTP/RTP Considerations

TFTPv2 (RFC 1350) provide an excellent yet simply method for transferring data between devices however given its synchronous operation, file transfers with small block sizes (512 octets) over large latency networks may result in slow data transfer rates. To improve the throughput performance, it is recommended to use higher order block sizes (4096 octets +) allowed for under RFC 2348. In doing so an implementer shall be aware of packet size limitations of UDP over error prone networks, there in, it is advisable to segment these blocks into manageable octets fragments and encapsulate them in an RTP frame and allow the remote RTP buffer reconstitute the block prior to processing. This will greatly increase the success rate of large block transfers and hence improve overall data throughput. In this case the RTP frame shall have a dynamic payload type and the standard TFTP data segment shall be encapsulated in the RTP payload.

4.2 Open Logical Channel Considerations

RFC 1350 in conjunction with RFC 2349 provide sufficient header information in the RRQ or WRQ message for the purpose of successfully identifying the type of transfer, the filename and the size of the proposed transferred file. An implementer may choose to also include this relevant information within the Open Logical Channel and Open Logical Channel Acknowledge message for the purpose of display and/or approval prior to opening the TFTP session.

The following procedure describes the optional method to convey relevant TFTP header information within the **OLC** and **OLCack** prior to opening the TFTP session.

When initiating a TFTP session, the **OLC** or where required the **OLCack** contains the **genericInformation** field which consists of a GenericInformation element for each file requested or to be transmitted. This shall consist of a **messageIdentifier** as in accordance with Table 4 and **messageContent** as prescribed in Table 5. Each GenericInformation element in the **genericInformation** field of the **OLC** shall contain an incremental value in the **subMessageIdentifier** field starting at 1 to uniquely identify the files that are to be transmitted or received.

Table 4 – H.245 OLC Generic Message Identifier for File Listing

OID Identifier Value	OID Name
iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) packetizer(17090) h245 (1) tftp_rtp (2) filelist (1)	H.460.MM file list

Table 5 – H.245 Generic Message Content for File Listing

Identifier	Indication	Value Type
1	File Transfer Direction	Refer Table 6
2	File Name	Refer Table 7
3	File Size	Refer Table 8

Table 6 – OLC Parameter TFTP Direction

Parameter name	tftpDirection
Parameter description	This field indicates the direction of the file Transfer. 1- Indicates a WRQ (send to remote host) 2- Indicates a RRQ (request from a remote host)
Parameter identifier value	1
Parameter status	Optional.
Parameter type	unsignedMin
Supersedes	–

Each OLC may contain information on the type of TFTP transfer. A **tftpDirection** field value of 1 shall indicate that the intention of opening the file transfer session for the purpose of reading a file off a remote host. A value of 2 indicates that the intention is to send a file.

Table 7 – OLC Parameter TFTP Filename

Parameter name	tftpFilename
Parameter description	This field indicates the filename to transfer
Parameter identifier value	2
Parameter status	Optional.
Parameter type	octetString
Supersedes	–

The tftpFilename field contains the filename requested or proposed file to be sent. This information can be used to check if the requested file is available for transfer or to check that the proposed file already exists.

Table 8 – OLC/OLCack Parameter TFTP File size

Parameter name	tftpFilesize
Parameter description	This field indicates the size in octets of the file to transfer
Parameter identifier value	3
Parameter status	Optional.
Parameter type	unsigned32Max
Supersedes	–

The **tftpFilesize** field may be conveyed within the OLC or the OLCack message depending on the direction of file transfer. Where the file is to be sent, that is with a **tftpDirection** value set of 1, the **tftpFilesize** message is contained in the **OLC** to indicate to the remote endpoint the size of the proposed file transfer. Where the file is requested from a remote endpoint then the **tftpFilesize** message is to be contained in the responding OLC or OLCack to notify the local endpoint of the size of the file being sent. Where multiple files are to be sent each **genericInformation** field shall contain

the **subMessageIdentifier** assigned to the file in the initial OLC. This information can then be used to determine if there is sufficient file system space to receive the file and also for display purposes to notify the user of the progress of the transfer.

4.3 NAT Traversal and Connectivity considerations and deviation from RFC1350

To ensure connectivity, an implementer may choose to probe for bi-directional connectivity prior to initiating a TFTP RRQ or WRQ request. To support this capability, this implementation deviates from RFC1350 to allow for the sending of probe packets (for this implementation allocated op code 0). Each endpoint upon opening ports may send numerous 2 byte TFPT packets containing op code of 0. Upon receipt of the probe packet at the remote endpoint, as with RRQ or WRQ packet as per RFC150, the remote endpoint shall respond with a TFTP packet with op code 4 ACK and block number value of 0. This shall indicate to the initiating endpoint that bi-directional connectivity has been achieved and a TFTP RRQ or WRQ request can be reliably sent.

Endpoints supporting this capability shall respond to a TFTP packet of op code 0 with an ACK (op code 4) packet with a **blocksize** value of 0.

Where upon segmentation is required for large TFTP blocks to be sent over several RTP packets and to allow the confirmation of the receipt of the entire block of data, this implementation proposes the use of a reply ERROR packet with an error code of 0 (undefined in RFC1350) to indicate to the sending endpoint that the packet received was incomplete and for the sending endpoint to resend the entire last TFTP block.

Endpoints supporting this capability shall interpret a TFTP ERROR messages with error code 0 as an indication of partial receipt of the block and to resend the entire last TFTP block.

Where a sending timeout is reached waiting for a response from the receiving endpoint after completing a TFTP block transmission, the sending endpoint shall resend the last TFTP Block. Where a receiving timeout is reached on the receiving endpoint, the endpoint shall not respond with any message and wait for the sending endpoint timeout to be reached and the retransmission of the TFTP Block.