International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.248.50

(07/2016)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Communication procedures

Gateway control protocol: NAT traversal toolkit packages

Recommendation ITU-T H.248.50

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	**H.240–H.259**
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Telepresence	H.420–H.429
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789
E-HEALTH MULTIMEDIA SERVICES AND APPLICATIONS	
Personal health systems	H.810–H.819
Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)	H.820–H.859
Multimedia e-health data exchange services	H.860–H.869

For further details, please refer to the list of ITU-T Recommendations.

Revised Recommendation ITU-T H.248.50

Gateway control protocol: NAT traversal toolkit packages

Summary

Recommendation ITU-T H.248.50 contains a series of ITU-T H.248 packages that enable various network address translator (NAT) traversal techniques to be employed in order to facilitate media flow between networks or user equipment and network-side located media gateways. Any of these packages may be utilized in any order to gather and map addresses, as well as maintain connectivity with and through NATs.

This revision of Recommendation ITU-T H.248.50 adds clarifications and further capabilities, such as:

– a new ITU-T H.248 session traversal utilities for NAT (STUN) consent freshness package;

– package-independent ITU-T H.248 procedures for specific NAT traversal use cases;

– interactive connectivity establishment (ICE) variants (full vs ICE lite; user equipment (UE)-embedded vs gateway-embedded ICE clients; vanilla vs trickle ICE; ICE for user datagram protocol (UDP) vs ICE for TCP; ICE for single-homed vs multi-homed host entities); and

– updates to keep alive support.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.248.50	2010-09-13	16	11.1002/1000/10984
1.1	ITU-T H.248.50 (2010) Cor. 1	2012-02-13	16	11.1002/1000/11542
2.0	ITU-T H.248.50	2016-07-14	16	11.1002/1000/12919

Keywords

FW, Gateway, ITU-T H.248, ICE, NAT, STUN, TURN.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T H.248.50

Gateway control protocol: NAT traversal toolkit packages

1 Scope

This Recommendation describes packages to enable various network address translator (NAT) traversal techniques to be employed in order to facilitate media flow between networks. The media gateway controller (MGC) may utilize any of the packages in any order to gather addresses, map them and then maintain connectivity with and through NATs.

The packages described in this Recommendation allow an ITU-T H.248 MGC and media gateway (MG) to use the techniques defined by:

– simple session traversal utilities for NAT (STUN) reflexive address mapping as defined in [IETF RFC 3489] and [IETF RFC 5389];

– relayed address mapping using the traversal using relays around NAT (TURN) techniques as described in [IETF RFC 5766];

– comprehensive NAT traversal interactive connectivity establishment (ICE) techniques as described in [IETF RFC 5389].

In order to maintain backward compatibility, packages have been produced for both STUN as defined by [IETF RFC 3489] and by [IETF RFC 5389].

Throughout this Recommendation it is assumed that the media gateway performs STUN server discovery through the use of domain name system (DNS) lookup.

Figure 1 summarizes the various packages as defined by the initial Recommendation. Every package is self-contained and does not use the extension principle.

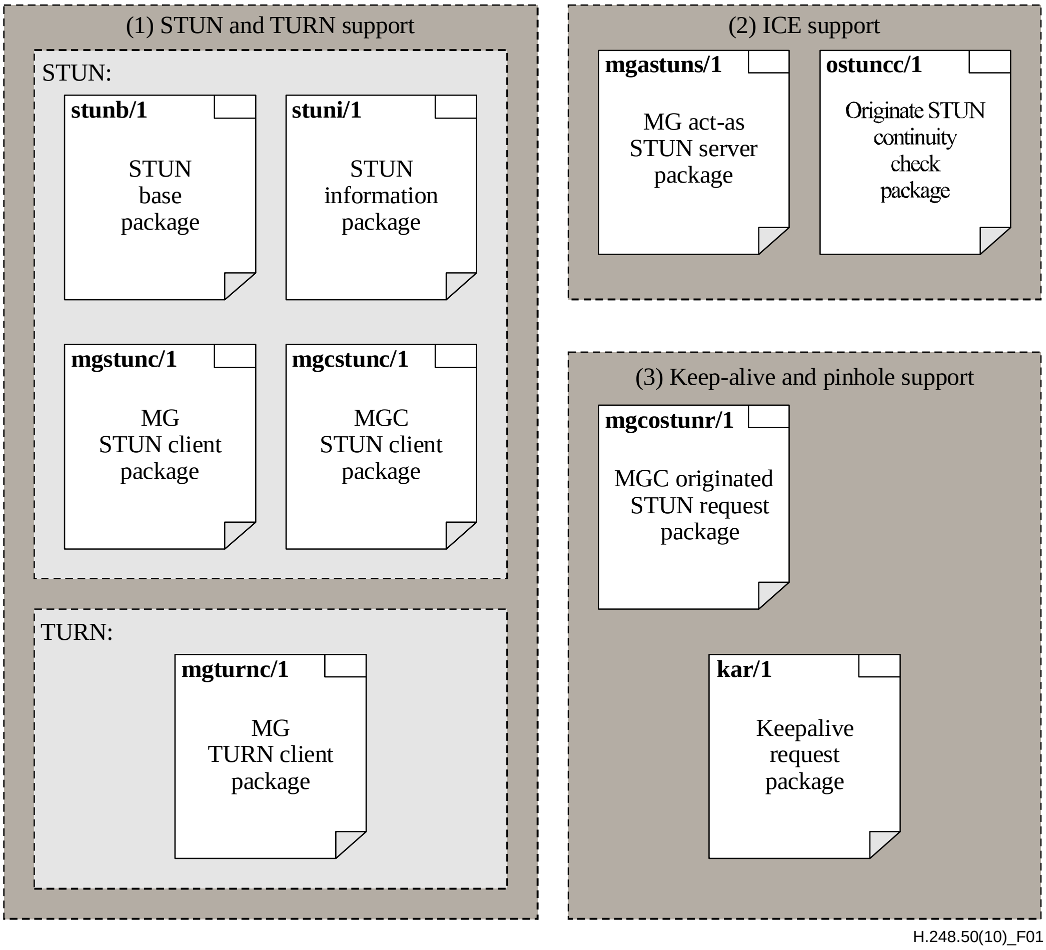


Figure 1 – Landscape of NAT traversal toolkit packages, categorized into three application areas (status of ITU-T H.248.50 (09/2010))

ITU-T H.248.50 (07/2016) adds clarifications and further capabilities, such as:

– a new ITU-T H.248 *STUN consent freshness* package;

– package-independent ITU-T H.248 procedures for specific NAT traversal use cases;

– ICE variants (full vs ICE lite; user equipment (UE)-embedded vs gateway-embedded ICE clients; vanilla vs trickle ICE; ICE for user datagram protocol (UDP) vs ICE for TCP; ICE for single-homed vs multi-homed host entities);

– updates to keep alive support; and

– clarifies support of "ICE restart" procedures.

1.1 Relation to other ITU-T H.248 related NAT traversal mechanisms

The following covers the applicability of this Recommendation versus other ITU-T H.248 supported NAT-T mechanisms:

– Transport protocol generic NAT-T method "media latching" [ITU-T H.248.37]: this is an orthogonal mechanism and may be used in combination with this Recommendation; some of the mechanisms in this Recommendation make "media latching" unnecessary.

NOTE – There is also a bearer plane latching process in the context of ICE, see clause 10.1.5 "STUN-specific media latching". The difference between both latching variants is illustrated in Appendix I. It is further noted that the IETF "media latching" function of [b-IETF RFC 7362] is synonymous to "H.248.37 latching" (because it makes the assumption that ICE/STUN/TURN is not applied for NAT traversal).

– Transport protocol specific NAT-T method "TCP merge mode" [b-ITU-T H.248.84]: this represents a TCP simultaneous open from an end-to-end perspective, which is, if applicable, used before any "ICE for TCP" procedures are selected.

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.

[ITU-T H.248.1] Recommendation ITU-T H.248.1 (2005), *Gateway control protocol: Version 3*.

[ITU-T H.248.14] Recommendation ITU-T H.248.14 (2009), *Gateway control protocol: Inactivity timer package*.

[ITU-T H.248.37] Recommendation ITU-T H.248.37 (2008), *Gateway control protocol: IP NAPT traversal package*.

[ITU-T H.248.40] Recommendation ITU-T H.248.40 (2013), *Gateway control protocol: Application data inactivity detection package*.

[ITU-T H.248.90] Recommendation ITU-T H.248.90 (2014), *Gateway control protocol: ITU-T H.248 packages for control of transport security using transport layer security (TLS).*

[ITU-T H.248.97] Recommendation ITU-T H.248.97 (2016), *Gateway Control Protocol: ITU-T H.248 support for control of SCTP bearer connections*.[IETF RFC 3489] IETF RFC 3489 (2003), *STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).*

[IETF RFC 3556] IETF RFC 3556 (2003), *Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth*.

[IETF RFC 3605] IETF RFC 3605 (2003), *Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP).*

[IETF RFC 4566] IETF RFC 4566 (2006), *SDP: Session Description Protocol*.

[IETF RFC 4787] IETF RFC 4787 (2007), *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*.

[IETF RFC 5245] IETF RFC 5245 (2010), *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols*.

[IETF RFC 5389] IETF RFC 5389 (2008), *Session Traversal Utilities for NAT (STUN).*

[IETF RFC 5766] IETF RFC 5766 (2010), *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).*

[IETF RFC 5780] IETF RFC 5780 (2010), *NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN).*

[IETF RFC 6544] IETF RFC 6544 (2012), *TCP Candidates with Interactive Connectivity Establishment (ICE).*

[IETF RFC 6888] IETF RFC 6888 (2013), *Common Requirements for Carrier-Grade NATs (CGNs).*

[IETF RFC 7675] IETF RFC 7675 (2015), *Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 full trickle ICE mode [b-IETF trickle-ice]: Regular mode of operation for *trickle ICE* agents, used in opposition to the *half trickle mode* of operation.

3.1.2 half trickle ICE mode [b-IETF trickle-ice]: A *trickle ICE* mode of operation where the *offerer* gathers its first generation of candidates strictly before creating and sending the offer. Once sent, that offer can be processed by *vanilla ICE* agents and does not require support for [b-IETF trickle-ice]. It also allows *trickle ICE* capable *answerers* to still gather candidates and perform connectivity checks in a non-blocking way, thus roughly offering "half" the advantages of trickle ICE. The mechanism is mostly meant for use in cases where support for trickle ICE cannot be confirmed prior to sending a first offer.

3.1.3 NAT traversal [b-ITU-T Y.2111]: The operation of adapting the IP addresses so that the packets in the media flow can pass through a far-end (remote) NAT.

3.1.4 network address translation [b-ITU-T Y.2111]: The operation by which IP addresses are translated (mapped) from one address domain to another address domain.

3.1.5 pinhole [ITU-T H.248.37]: A configuration of two associated H.248 IP terminations within the same context, which allows/prohibits unidirectional forwarding of IP packets under specified conditions (e.g., address tuple).

3.1.6 STUN agent [IETF RFC 5389]: An entity that implements the STUN protocol. The entity can be either a *STUN client* or a *STUN server*.

3.1.7 STUN client [IETF RFC 5389]: An entity that sends STUN requests and receives STUN responses. A *STUN client* can also send indications.

3.1.8 STUN consent freshness [IETF RFC 7675]: Maintaining and renewing *consent* over time. *Consent* relates to the mechanism of obtaining permission from the remote endpoint to send non-ICE traffic to a remote transport address. Consent is obtained using ICE. Note that this is an application-level consent; no human intervention is involved.

3.1.9 STUN server [IETF RFC 5389]: An entity that receives STUN requests and sends STUN responses. A *STUN server* can also send indications.

3.1.10 symmetric NAT [IETF RFC 3489]: A symmetric NAT is one where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port.

3.1.11 vanilla ICE [b-IETF trickle-ice]: The Interactive Connectivity Establishment protocol as defined in [IETF RFC 5245].

3.2 Terms defined in this Recommendation

3.2.1 ICE-full (mode, implementation): The ICE protocol capability set that performs the complete set of ICE requirements according to section 4.1 of [IETF RFC 5245].

NOTE – This definition is based on section 3 of [IETF RFC 5245]: "*Full: An ICE implementation that performs the complete set of functionality defined by this specification.*" The RFC uses the term "full ICE" to reference this functionality.

3.2.2 ICE-lite (mode, implementation): The ICE protocol capability set that performs the subset of ICE requirements according to section 4.2 of [IETF RFC 5245].

NOTE – This definition is based on section 3 of [IETF RFC 5245]: *"Lite: An ICE implementation that omits certain functions, implementing only as much as is necessary for a peer implementation that is full to gain the benefits of ICE. Lite implementations do not maintain any of the state machines and do not generate connectivity checks."*

3.2.3 trickle ICE: The extended ICE protocol capabilities, beyond *vanilla ICE*, as defined in [b-IETF trickle-ice]. Trickle ICE includes half trickle and full trickle ICE modes.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

L4	(Protocol) Layer 4 (= IP transport protocol layer)
L4+	Upper Layers versus L4
ABNF	Augmented Backus-Naur Form
B2BIH	Back-to-Back IP Host
B2BUA	Back-to-Back User Agent
CRLF	Carriage-Return and Line-Feed (sequences)
DCCP	Datagram Congestion Control Protocol
DNS	Domain Name System
DTLS	Datagram Transport Layer Security
FW	FireWall
GFO	Gateway Free Operation
ICE	Interactive Connectivity Establishment
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Services Digital Network
IUA	ISDN User Adaptation
LD	Local Descriptor
MG	Media Gateway
MGC	Media Gateway Controller
NAT	Network Address Translation (or Translator)
NAT-T	Network Address Translation and Traversal
O/A	Offer/Answer
PES	PSTN/ISDN Emulation Subsystem
PSTN	Public Switched Telephone Network
RD	Remote Descriptor
RMG	Residential Media Gateway
RMGC	Residential Media Gateway Controller
RTCP	RTP Control Protocol

RTO	Retransmission TimeOut [IETF RFC 5389]
RTP	Real Time Protocol
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SEP	Stream Endpoint
SIP	Session Initiation Protocol
SRTP	Secure RTP
STUN	Session Traversal Utilities for NAT
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TURN	Traversal Using Relays around NAT
UDP	User Datagram Protocol
UA	User Agent
UE	User Equipment
UNSAF	UNilateral Self-Address Fixing
VPN	Virtual Private Network
WebRTC	Web Real-Time Communication

5 Conventions

None.

6 Toolkit usage

6.1 ITU-T H.248.50 usage in different network models

The ITU-T H.248 packages of this Recommendation may be applied in various network configurations. The clauses below illustrate some main scenarios.

6.1.1 ITU-T H.248 MGC/MG as interim nodes in the end-to-end path between user equipment (with STUN client/server and/or ICE support)

The NAT traversal support protocols: STUN, TURN and ICE are IP application protocols. They are fundamentally deployed in IP hosts (behind NAT device(s)) and IP network servers. The IP host function is located in the user equipment (UE) in Figure 2. The end-to-end IP bearer-path goes through an ITU-T H.248 MG (with an ITU-T H.248 IP-IP Context). There may be multiple NAT devices in the IP bearer-path (and IP signalling path(s)). Figure 2 shows an example with four IP realms, separated by two standalone NAT devices. The third NAT function is provided by the ITU-T H.248 MG.

NOTE – The MG-internal NAT function is realized by the so-called back-to-back IP host (B2BIH) mode.

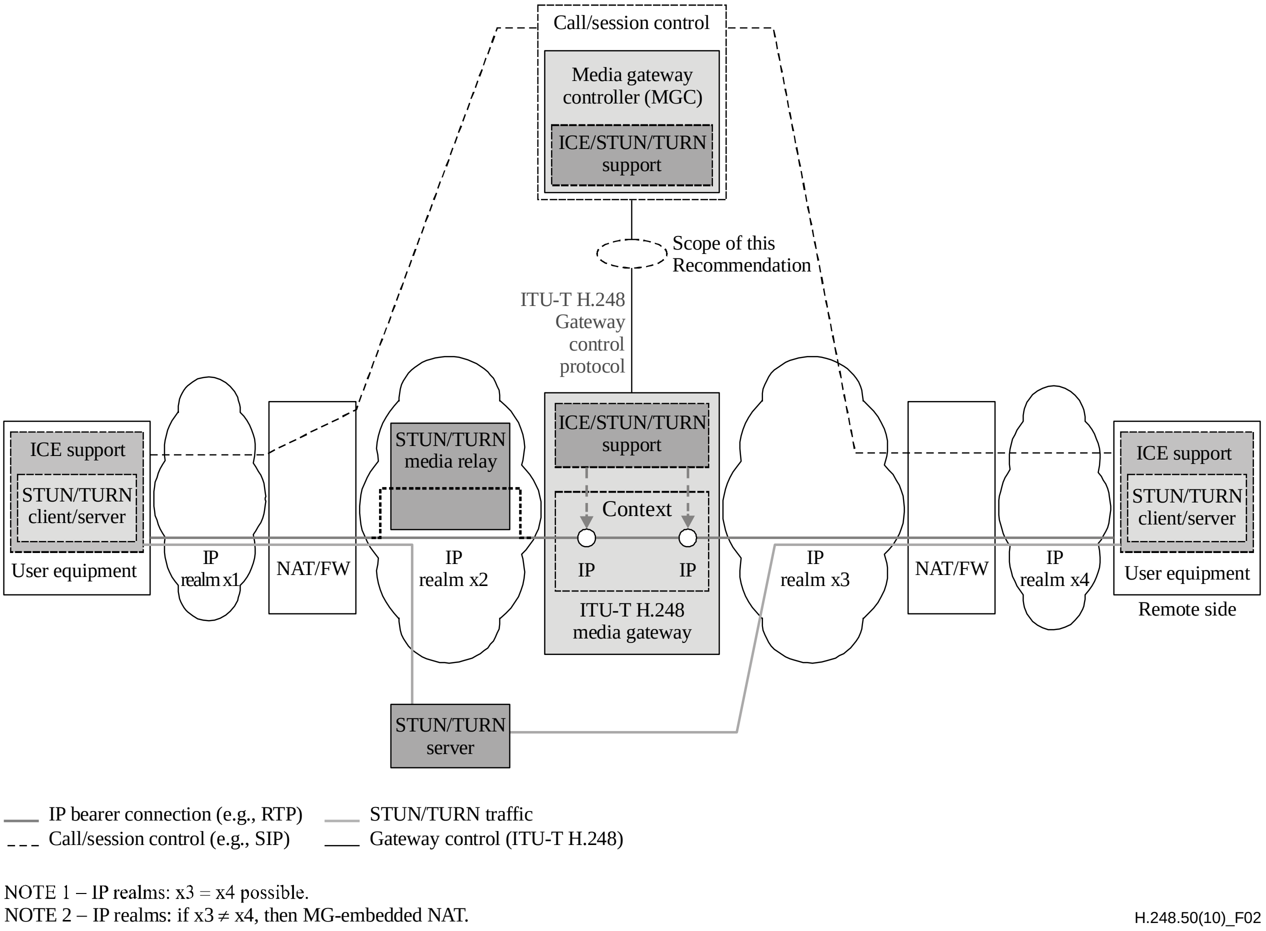


Figure 2 – Network model – ITU-T H.248 MGC/MG as interim nodes in the end-to-end path between user equipment (with STUN client/server and/or ICE support)

The ITU-T H.248 gateway (MGC and MG entity) may be requested to support particular STUN/TURN/ICE scenarios. For instance, the ITU-T H.248 gateway may be requested to act in a "proxy role" of the STUN client function (e.g., in case that the MG and UE would be both located in the same IP realm and when the UE could not provide the STUN client function itself).

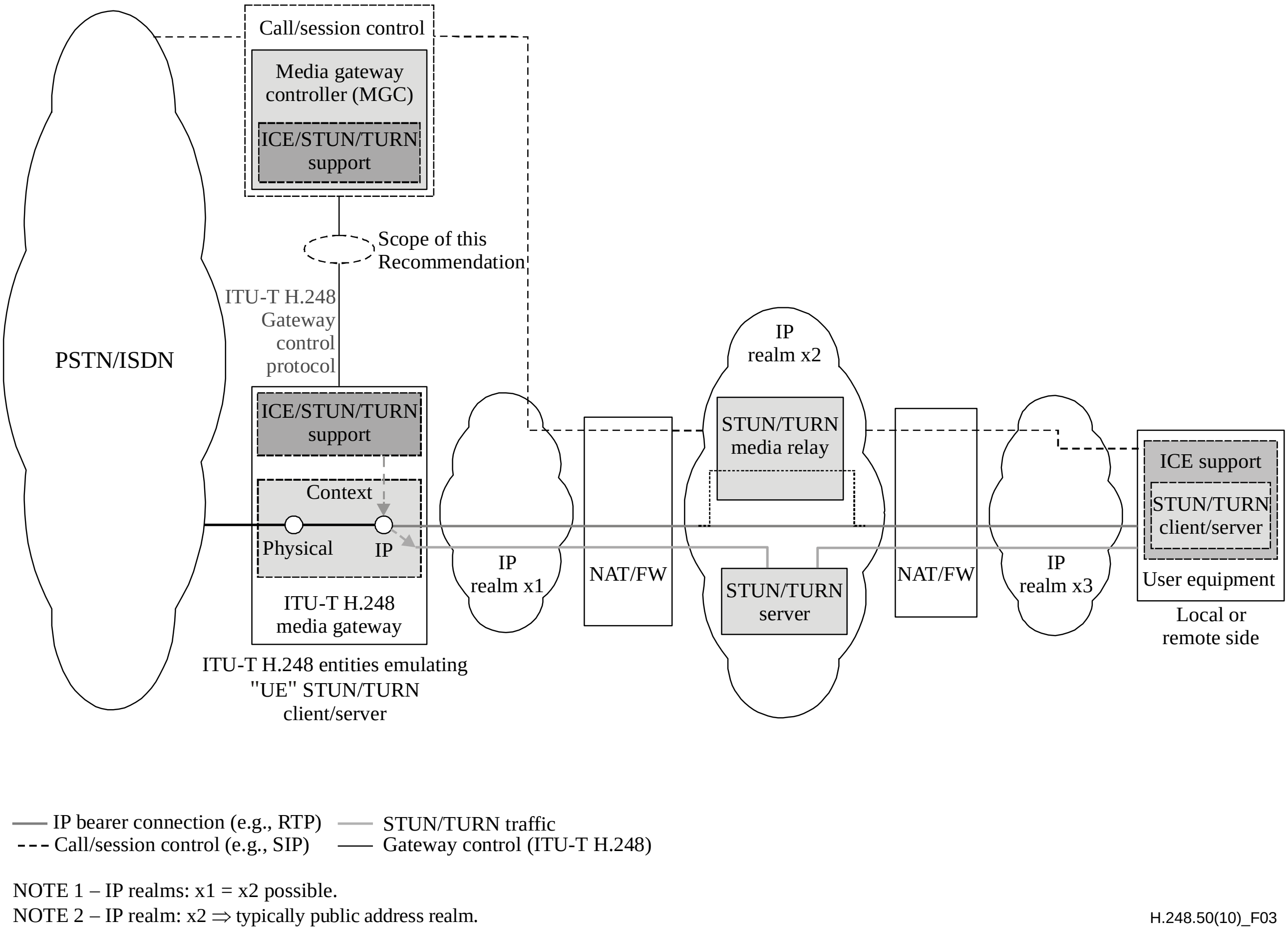
It may be noted that the relation between the assist protocols (STUN, TURN and ICE) and the call/session control protocol (like session initiation protocol (SIP)) may vary between a loose and a tighter coupling mode, e.g., some examples with regard to the time relation:

– timely tightly coupled: e.g., the ICE address gathering phase and subsequent SIP INVITE phase (before connectivity check phase);
– timely loosely coupled: e.g., the STUN keep pinhole open mechanism and NAT pinhole timer settings (which may vary between 30 seconds and the timescale of minutes and hours);
– timely de-coupled: e.g., the basic STUN mechanism for analysing the mode of operation(s) of the installed base of NAT devices.

The function of the last bullet item may be decoupled from call/session control phases and done in advance (because network topology and NAT behaviour changes rather slowly). The ITU-T H.248 gateway may be used for a NAT traversal support function on each time-scale.

6.1.2 ITU-T H.248 MGC/MG emulates a user agent (with STUN client/server and/or ICE support)

The evolution from the legacy public switched telephone network/integrated services digital network (PSTN/ISDN) to IP-based networks like the PSTN emulation subsystem (PES) or the IP multimedia subsystem (IMS) may lead to a network model as depicted in Figure 3. The ITU-T H.248 gateway may be located at the border between the circuit-switched and IP networks. The ITU-T H.248 gateway may be requested to emulate the behaviour of IP user equipment with regard to the NAT traversal support function.



Call/session control
Media gateway controller (MGC)
ICE/STUN/TURN support
Scope of this Recommendation
ITU-T H.248 Gateway control protocol
PSTN/ISDN
ICE/STUN/TURN support
Context
Physical
IP
ITU-T H.248 media gateway
ITU-T H.248 entities emulating "UE" STUN/TURN client/server
IP realm x1
NAT/FW
IP realm x2
STUN/TURN media relay
STUN/TURN server
NAT/FW
IP realm x3
ICE support
STUN/TURN client/server
User equipment
Local or remote side
IP bearer connection (e.g., RTP)
Call/session control (e.g., SIP)
STUN/TURN traffic
Gateway control (ITU-T H.248)
NOTE 1 – IP realms: x1 = x2 possible.
NOTE 2 – IP realm: x2 ⇒ typically public address realm.
H.248.50(10)_F03

Figure 3 – Network model – ITU-T H.248 MGC/MG emulates a user agent (with STUN client/server and/or ICE support)

It may be noted that the TURN media relay function is outside of the ITU-T H.248 gateway and provided by a separate network server in the previous two examples.

6.1.2.1 Scenario when MG and MGC located in different IP realms

Where the MGC and MG are in different IP realms, the ITU-T H.248 signalling and media flows may traverse the same NAT/firewall (FW) (see also Figure 4, using the example of a residential media gateway (RMG)). The techniques described in this Recommendation may be used for media flow NAT/FW traversal. Signalling (call/session) NAT traversal is generally not in the scope of this Recommendation, however ITU-T H.248 NAT traversal may benefit from the use of ITU-T H.248-based peer-to-peer polling mechanisms (like e.g., [ITU-T H.248.14] or empty audits on ITU-T H.248 level, or e.g., stream control transmission protocol (SCTP) indications on transport connection level) in order to maintain NAT bindings. Opening of the NAT binding for ITU-T H.248 traffic would be subject of the initial ServiceChange messages for MG registration.

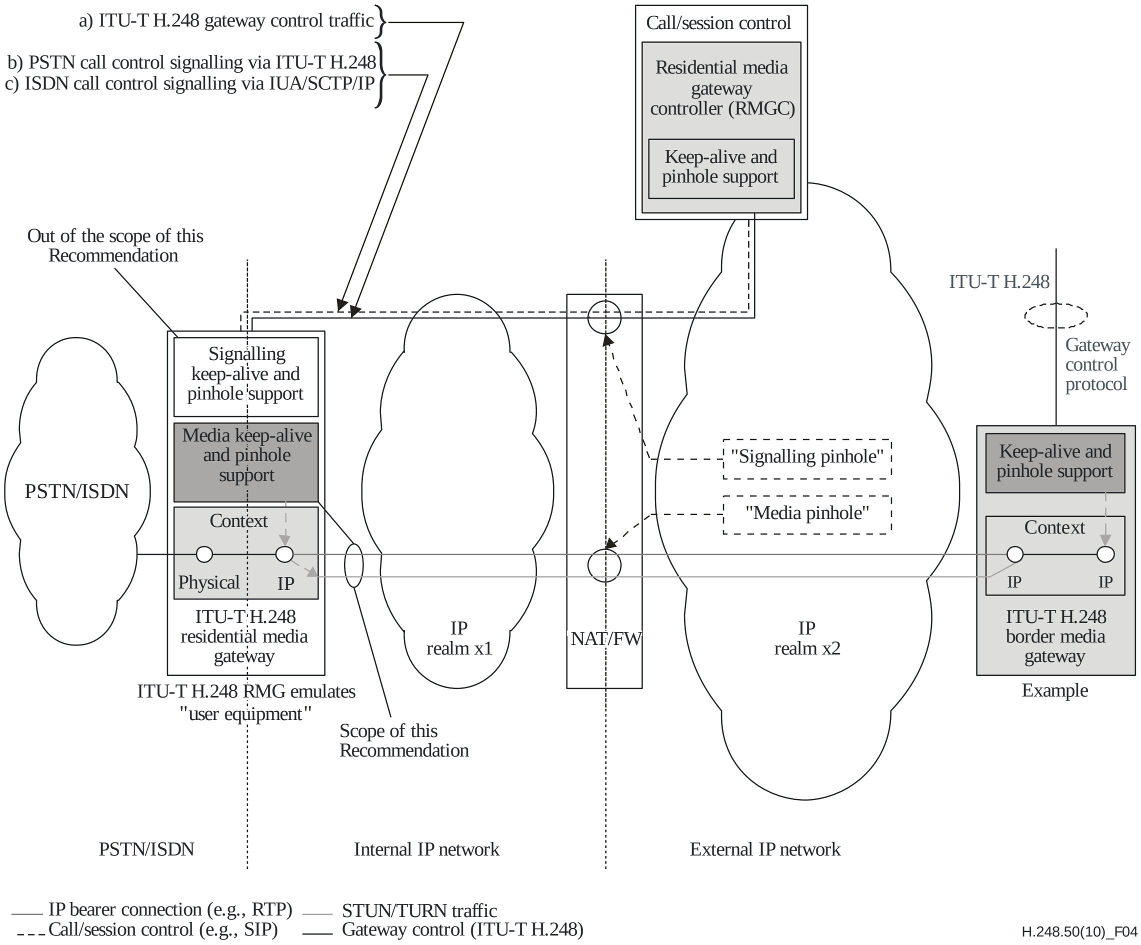
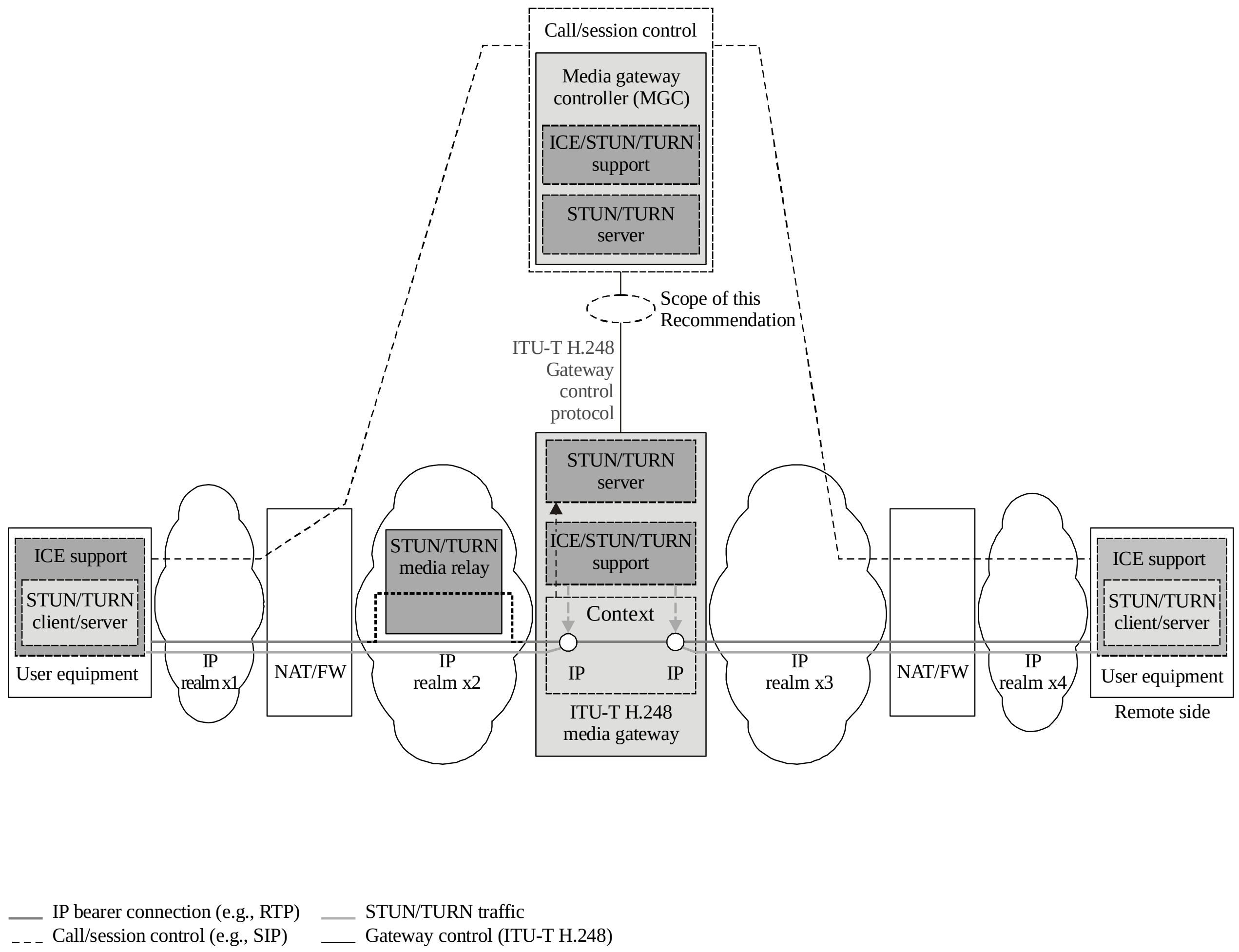

a) ITU-T H.248 gateway control traffic
b) PSTN call control signalling via ITU-T H.248
c) ISDN call control signalling via IUA/SCTP/IP
Call/session control
Residential media gateway controller (RMGC)
Keep-alive and pinhole support
Out of the scope of this Recommendation
ITU-T H.248
Gateway control protocol
Signalling keep-alive and pinhole support
Media keep-alive and pinhole support
PSTN/ISDN
Context
Physical
IP
"Signalling pinhole"
"Media pinhole"
Keep-alive and pinhole support
Context
IP
IP
ITU-T H.248 residential media gateway
IP realm x1
NAT/FW
IP realm x2
ITU-T H.248 border media gateway
Example
ITU-T H.248 RMG emulates "user equipment"
Scope of this Recommendation
PSTN/ISDN
Internal IP network
External IP network
IP bearer connection (e.g., RTP)
Call/session control (e.g., SIP)
STUN/TURN traffic
Gateway control (ITU-T H.248)
H.248.50(10)_F04

Figure 4 – Network model – Scenario when MG and MGC are located in different IP realms – Keep-alive and pinhole support

Keep "pinhole open" mechanisms for media IP flows are in scope of this Recommendation. Keep "pinhole open" mechanisms for signalling IP flows are out of scope of this Recommendation; such methods may be e.g., addressed by correspondent ITU-T H.248 profile specifications for such ITU-T H.248 gateways.

6.1.3 ITU-T H.248 MGC/MG provides STUN/TURN server functionality

The STUN/TURN server function could be principally embedded in a MG, MGC or gateway (MGC/MG tandem), see Figure 5.



NOTE 1 – IP realms: x3 = x4 possible.
NOTE 2 – IP realms: if x3 ≠ x4, then MG-embedded NAT.

H.248.50(10)_F05

Figure 5 – Network model – ITU-T H.248 MGC/MG provides STUN/TURN server functionality

The STUN server function requires the processing of incoming STUN Binding Request messages and the reply by correspondent STUN Binding Response messages. In case of NAT devices in "symmetric NAT" mode, additional support by TURN is required. The TURN server function requires the processing of TURN messages (Allocate Request, Allocate Response and Send Request).

The STUN server and TURN server functions may be provided by an ITU-T H.248 gateway on a MG or MGC level (as indicated in Figure 5). However, the TURN media relay function should be out of scope of MGC nodes (see clause 6.1.4).

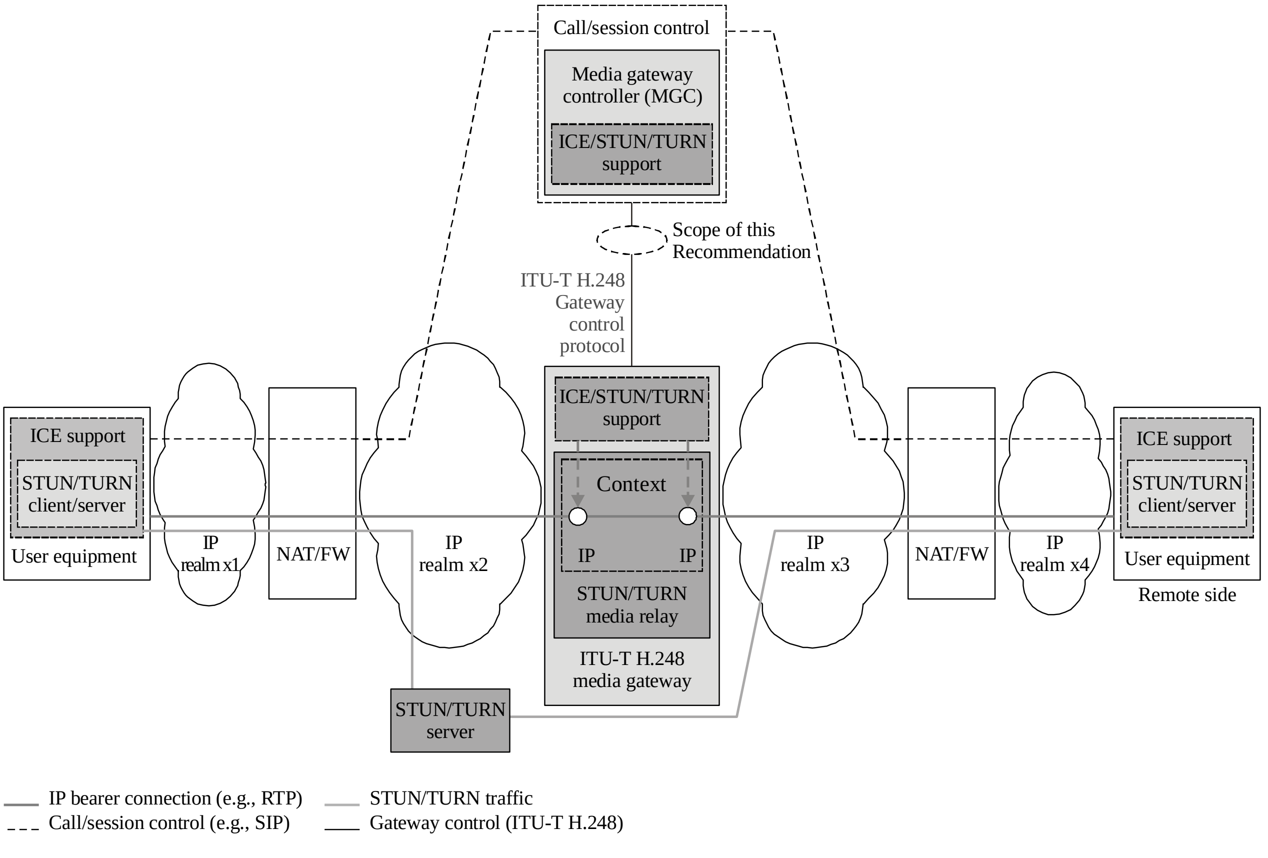
The advantage of a STUN/TURN server at MGC level is the close location to call/session control. For instance, it may be beneficial to have information available concerning interim NAT devices and their behaviour, NAT binding lifetime information (e.g., a TURN attribute) or IP address usage. A close location of possible SIP server function (like a SIP proxy, application level gateway or back-to-back user agent (B2BUA) function) and STUN/TURN servers may also be beneficial.

The advantage of a STUN/TURN server at MG level may be e.g., driven by a functional sharing performance model for off-loading the MGC node from STUN/TURN message processing functions, or e.g., the coupling of a TURN server and TURN media relay in an ITU-T H.248 MG node.

6.1.4 ITU-T H.248 MG provides TURN media relay functionality

The TURN media relay function may be embedded in a MG itself, see Figure 6. The TURN media relay function is inserted in the end-to-end IP bearer-path. The decision for routing the IP bearer connection over a TURN media relay is either already known before call/session establishment, or done during that phase due to a call/session-driven STUN process. The TURN media relay function may require substantial resources for processing and forwarding IP bearer path packets. This is the primary reason for excluding such a function from MGC entities.

There are on the other side many good reasons for combining the TURN media relay function and ITU-T H.248 MG function. For example, this is because the MG may inherently provide the TURN media relay NAT function, or because of a "simpler" IP bearer-path routing process, or due to QoS reasons (e.g., the native TURN media relay provides only coarse support for bearer-path resource reservation by the single attribute "Bandwidth", but is lacking support for session-dependent policing functions per se), or the ability of the MGC to access TURN media relay server information.



NOTE 1 – IP realms: x3 = x4 possible.
NOTE 2 – IP realms: if x3 ≠ x4, then MG-embedded NAT.

Figure 6 – Network model – ITU-T H.248 MG provides TURN media relay functionality

6.1.5 "STUN server" capability

A network domain may provide one or multiple STUN server(s). A STUN server may be realized as a non-ITU-T H.248 network element or as ITU-T H.248 entity-embedded STUN server functions (in context of this Recommendation). For instance, the STUN agent provided by the MG is then called "MG act-as STUN server" function.

The STUN service(s) may be provided by a single or multiple STUN servers; typically, dependent on the network configuration and/or STUN application.

The internal modeling of the STUN protocol in "service categories" (protocol functions) is difficult; however, STUN server functions could be classified in *call-independent* and *call-dependent* categories. It should be noted that such a classification scheme is not useful in general, but suitable in the context of ITU-T H.248 gateways.

The call-dependent category relates to the ICE connectivity check and has to be performed by the entity in the call/session control path that terminates ICE, either the ICE end point located in a terminal or a gateway, (e.g., a session border gateway). Providing a single STUN server in the network domain thus implies that call control signalling corresponding to all call/sessions using ICE has to be routed through that gateway (such as in Figure 5), unless the end point itself provides the server functionality.

An example of STUN server function distributions for STUN application 'ICE' according to the network model in Figure 2 (see also description of the scenario in clause 6.1.1 or Appendix I.1):

A stand-alone "STUN server" is involved during the STUN-based address gathering phase. And the ITU-T H.248 MG as "STUN server" is involved during the STUN-based connectivity check phase. The first STUN server function is "call-independent" because it actually occurs before any call control signalling (from MGC perspective), and the second STUN server function is "call-dependent" because associated with call control signalling.

The STUN server involved during STUN-based address gathering phase could also be provided by an ITU-T H.248 MG entity (such as in case of Figure 5).

6.1.6 ICE capability set: ICE full vs ICE lite

ICE allows two explicit modes of operations, where "ICE full" represents the complete capability set and "ICE lite" a subset of "ICE full". [IETF RFC 5245] defines an explicit signalling element (session description protocol (SDP) "a=ice-lite" attribute) for ICE mode support indication. The omission / inclusion of this SDP attribute in SIP signalling indicates ICE full / lite support. However, [IETF RFC 5245] does not provide a detailed description of the behaviour of the ICE lite mode; the information is rather distributed across the entire ICE RFC.

NOTE – The rationale behind this is the expectation that ICE lite is only a temporary implementation step which should evolve to ICE full support. This assumption might be valid for terminal-embedded ICE agents, but not necessarily valid for gateway-embedded ICE support as in scope of this Recommendation.

Appendix III describes the differences between the two ICE modes at the level of detail of interest to ITU-T H.248 gateways.

6.1.7 Principal network models concerning location of ICE endpoints

ICE represents a single algorithm, distributed over two ICE agents in a network. There are two main models concerning the mapping of the two ICE agents to the network topology (Table 1):

This aspect should be taken into account when looking at the use cases in clauses 6.1.1 to 6.1.4.

Table 1 – Network models concerning location of ICE endpoints

Model	Characteristics	Comments
Terminal-to-terminal (peer-to-peer)	Both ICE endpoints are located in terminal embedded IP host entities	This is the base model of IETF for the design of ICE. All IETF "ICE RFCs" use this model.
Terminal-to-network	One ICE endpoint is located in IP network infrastructure equipment, such as an ITU-T H.248 IP-to-IP media gateway (acting as B2BIH or IP router) within the end-to-end IP media path. This leads to following consequences: • The end-to-end IP media path is divided (by the ITU-T H.248 MG) in two segments: 1) "ICE segment"; and 2) "another non-ICE based segment". • Special case: both segments use an ICE-based NAT traversal independent of each other.	Major difference to the IETF base model: the MG might not be "located behind a NAT" (from the perspective of ICE), and thus be always reachable ("full IP connectivity without ICE-based NAT traversal support").

6.2 Overview of toolkit NAT traversal techniques

This Recommendation describes the various mechanisms used in order to secure media flow traversal of networks where NATs are used. In essence, the packages described provide a toolkit for the MGC to use in order to: gather a list of potential NAT mapped address (address gathering), check connectivity associated with those addresses, and maintain the connectivity associated with those addresses. The Recommendation is structured around the various techniques that have been defined:

1) STUN usage according to [IETF RFC 3489];

2) STUN usage according to [IETF RFC 5389];

3) TURN usage according to [IETF RFC 5766];

4) ICE techniques according to [IETF RFC 5245].

NOTE – At the time of publication of this Recommendation, this RFC was being updated by [b-IETF RFC 5245bis] and [b-IETF ice-sip-sdp]. These updates are out of scope of this Recommendation.

5) Other techniques used to maintain connectivity.

These techniques have typically not considered the scenario where the media flows are initiated from a split MGC and MG.

6.3 ITU-T H.248 call/bearer separation, connection model and IP addresses for ephemeral terminations

The splitting of the so-called "user agent" functionality into a control and media components means that there needs to be coordination between the two components. As the MGC usually constructs call and session control messages, it must be able to request information from the MG in order to build these messages. For instance, it would need address and media information to place in an SDP offer.

Media gateways typically support several different source/destination IP address/port combinations per peer-to-peer media connection. This may be due to different media (e.g., audio or video), different single media capabilities (codec A versus codec B) and to handle events (e.g., real time protocol (RTP) control protocol (RTCP)). These capability sets may be advertised to a remote party which selects the appropriate media set to use. In order for media connectivity to be successful for any of the sets, any media stream that encounters a NAT should have its address mapped. At a call/session level, a correlation is given between the local "native" IP address/ports of the MG and the mapped addresses. Such a scheme is detailed in section 4.3 of [IETF RFC 5245]. The use of ITU-T H.248 between a MGC and MG introduces a connection model that needs to be considered when correlating local "native" addresses and mapped addresses using ICE techniques. Furthermore, some NAT traversal techniques do not use ICE, thus this particular candidate scheme is not appropriate for them.

As such in this Recommendation, where ICE is not used, the correlation is in the form of an ordered list of values, where:

The first position on the list corresponds to the first address in the first group associated with a particular stream. The second position is associated with either a second address in the first group or, if there are no further addresses, it is associated with the first address in the second group. The parameter "Address Correlation" in the Base STUN package describes the mapping between the list position and the place in the Local Descriptor (LD).

6.4 Specific SDP information elements

6.4.1 SDP for vanilla ICE

6.4.1.1 UDP-based address candidates

Where ICE is used, the SDP CANDIDATE attribute [IETF RFC 5245] is used.

The so-called "candidate scheme" does not alter the actual media flows. That is, a MG will use the source address in the Local Descriptor to send from, and the destination address in the Remote Descriptor (RD) to send to. In the text encoding, this is the SDP connection address (c=) and media (m=) lines. Therefore, if the MGC chooses a candidate other than the local address, this shall be reflected in the Local and Remote Descriptors. This may have implications on how the media is sent. For example, if this address is a relay address, then the data may have to be sent using TURN send indications.

6.4.1.2 TCP-based address candidates

Where ICE is used for TCP [IETF RFC 6544] (see also Appendix II), the SDP CANDIDATE attribute from [IETF RFC 5245] is used with its grammar extended according to section 4.5 of [IETF RFC 6544].

6.4.2 SDP for extended ICE

6.4.2.1 Trickle ICE

Appendix IV provides some background information on trickle ICE, from the perspective of ITU-T H.248 gateways.

Where *trickle ICE* is used, the SIP-based call control signalling (according to [b-IETF trickle-ice] and [b-IETF trickle-sip]) uses the following signalling extensions in brief:

1. start of call establishment phase: indication of trickle ICE support via SDP attribute "a=ice-options:trickle";
2. trickling phase: signalling of candidate updates via SIP INFO messages; and
3. stop of trickling phase: announcing end of candidates via SDP attribute " a=end-of-candidates".

The three trickle ICE phases affect the ITU-T H.248 MGCs' SIP-based call control signalling interface only. There are not necessarily correspondent impacts on the ITU-T H.248 gateway control signalling. This seems to be rather conditional, dependent on the network configuration and IP bearer path routes. A detailed analysis of trickle ICE is for further studies.

6.5 Overview of NAT traversal support mechanisms (by ITU-T H.248 entities)

6.5.1 Address latching support

Legacy "media latching" as introduced for network infrastructures without ICE/STUN/TURN (see also [b-IETF RFC 7362]) is covered in the scope of [ITU-T H.248.37]. Clause 10.1.5 describes a STUN-specific media latching function in the context of ICE, which effectively leads to the same autonomous adaptation of IP transport addresses behaviour in the IP bearer plane.

6.5.2 Basic STUN/TURN support (ICE-less)

See clause 7.

6.5.3 ICE-controlled STUN/TURN

See clause 8. The ICE methodology defines a "superior" protocol that uses the STUN/TURN mechanism and other protocols. The use of ICE is also tightly coupled with SIP and in particular with SIP/SDP offer/answer (O/A) procedures. ICE is executed between a pair of ICE agents. There are two device types concerning the embodiment of ICE agents with respect to their location in the IP network infrastructure:

– terminal-embedded ICE agent: user equipment with ICE agent function and SIP UA function;

– gateway-embedded ICE agent: a decomposed ITU-T H.248 gateway provides the ICE agent function in a distributed manner (with the SIP signalling function located at MGC level).

This clause describes the ICE process from the perspective of a generic ICE agent, i.e., it does not differentiate between the above two embodiments (but indicates possible ITU-T H.248 impacts for the gateway-provided ICE agent).

The execution of ICE may be basically divided into several consecutive phases:

1) Gather addresses

– Purpose: Address gathering.

– This phase may be based on STUN and/or TURN procedures as required.

– Used ITU-T H.248.50 packages: None.

2) Prioritizing addresses

– Purpose: optimize final IP bearer path (routing).

– Function provided either by

  a) MG, in case of a MGC request via SDP CANDIDATE attribute (MG needs to provide then the STUN server discovery, see clause 8); or

  b) MGC, in its capacity as "ICE endpoint" and SDP O/A processing; or

  c) external (SIP) server, in case of a network deployment model where the address gathering and prioritization functions are provided by non-ITU-T H.248 entities;

– Used ITU-T H.248.50 packages: SDP "a=candidate:" attribute and *ostuncc* package (because the STUN connectivity checks result in priority value assignments.

– Address prioritization rules: The basic rules for address prioritization are described in section 4.1.2.1 of [IETF RFC 5245]. The purpose of ICE multihomed and IPv4/IPv6 dual stack fairness support may require other prioritization algorithms, see [b-IETF ice-dualstack] and Appendix V. The applicable prioritization rules should be explicitly part of an ITU-T H.248 profile specification in case that an ITU-T H.248 entity is responsible for that ICE function.

3) Call/session control signalling: generate SDP Offer (for SIP signalling)

– Purpose: generate SDP Offer content (if originating side).

– Multiple SDP attributes may be involved: the basic ICE protocol introduces seven ICE-specific SDP attributes (see [IETF RFC 5245]).

NOTE 1 – The values of media-level ICE SDP attributes could be requested from the MGC and provided by the MG, whenever the MG is involved in the resource management of such information.

– Used ITU-T H.248.50 packages: None (due to application control protocol related function).

4) Call/session control signalling: Negotiations – Offering and Answering (initiate and accept SIP messages

– Purpose: Address advertisement.

– This phase is typically related to the initiation of a SIP INVITE (thus, out of scope of this Recommendation).

NOTE 2 – The peer user (called party, invited party) may also start a correspondent "address gathering" procedure (based on STUN and/or TURN as required). These remote procedures would not be visible for the local MGC/MG entities.

– The peer (SIP) user may reply with a SIP OK (thus, out of scope of this Recommendation).

– Used ITU-T H.248.50 packages: None (due to application control protocol related function).

5) Connectivity checks

– This phase is again based on STUN and/or TURN procedures as required.

– The connectivity checks are based on the same 4-tuple(s) as (later) used for the ITU-T H.248 stream.

– The STUN/TURN messages may be thus multiplexed into the RTP/RTCP packet flow of the checked ITU-T H.248 stream (of an ITU-T H.248 RTP termination).

– Used ITU-T H.248.50 packages: mgastuns, ostuncc.

6) Completing ICE procedures

– One agent assumes ICE *controlling* role, other agent assumes ICE *controlled* role.

– Table III.1, table section 6 lists additional functions as part of the ICE completion phase.

– Used ITU-T H.248.50 packages: None or indirectly because ICE completion is subject of call control signalling between the two ICE endpoints.

NOTE 3 – The ICE completion process could result in the conclusion that the routed IP bearer path could possibly finally bypass the MG instance, which was initially involved in the above ICE procedural steps. The initially created ITU-T H.248 Context would be then obsolete from an ICE perspective, which could result in MGC performing either of the following actions:

a) the deallocation of the MG (which may be termed as gateway free operation (GFO)), or

b) the ITU-T H.248 stream is enabled for a transparent forwarding mode of operation.

7) Bearer connection: check/selection of media streams

– Used ITU-T H.248.50 packages: None.

The outlined, high level ICE process is not detailed concerning possible differences from ICE mode perspective. The ICE mode impact on gateway-embedded ICE agents is described in Appendix III.

7 STUN and TURN support

The packages defined in this clause allow the use of STUN and TURN techniques without the need for support of ICE and associated SDP, e.g., Candidates are not used.

7.1 STUN base package

Package name: STUN Base

Package ID: stunb (0x00bd)

Description: This package describes the mapping between transport addresses defined in the ITU-T H.248 Local Descriptor and the list position when used for STUN purposes.

Version: 1

Extends: None

7.1.1 Properties

7.1.1.1 Address correlations

Property name: Address Correlation

Property ID: ac (0x0001)

Description: This property details the list position of each IP Address in the ITU-T H.248 Local Descriptor. This package may be used by the MGC to determine the mapping between local IP addresses and list positions for STUN processing purposes.

NOTE – It is not mandatory to implement this package as the MGC may determine the list positions given the logic below. However it provides a mechanism in order to check the list positions.

Type: List of String

Possible values: Each element in the list of string SHALL be type **`AddressCorrelation`** according to the following augmented Backus-Naur form (ABNF):

```
AddressCorrelation = Listposition "|"
       Groupnumber "|" Instance "|" ComponentID
```

Where:

List position is an integer. The first position of the list shall be 1 and sequentially rising by one.

Groupnumber is an integer and is the ITU-T H.248 group number.

Instance is an integer. The first instance of a media format (as per section 5.14 of [IETF RFC 4566]) <fmt> field in "m=" line in a particular group shall be 1. For each subsequent media format the instance shall be incremented by one.

ComponentID is the identifier of a component. A component is a piece of a media flow requiring a single transport address. For RTP-based media flows, the RTP itself has a component ID of 1, and RTCP has a component ID of 2. For non RTP-based media flow, the component ID is 1.

The list SHALL contain an element for each IP address in the Local Descriptor.

For example, the values:

```
"1|1|1|1"
"2|1|1|2"
"3|1|2|1"
"4|1|2|2"
"5|2|1|1"
"6|2|1|2"
```

Would relate to the following SDP in the Local Descriptor:

```
v=0
c=IN IP4 192.168.1.100
m=audio 10000 RTP/AVP 4 18
v=0
c=IN IP4 192.168.1.200
m=audio 20000 RTP/AVP 0
```

The values of ReserveValue and ReserveGroup are "on".

Default: Empty String if no IP addresses are present in the Local Descriptor.

Defined in: LocalControl

Characteristics: ReadOnly

7.1.2 Events

None.

7.1.3 Signals

None.

7.1.4 Statistics

None.

7.1.5 Error codes

None.

7.1.6 Procedures

In order for the MGC to determine in a timely manner the correlation between local IP address and the STUN processing list positions, the MGC shall perform a CHOOSE ($) on the "*stunb/ac*" property whenever an IP address is added/changed/deleted to/from the ITU-T H.248 Local Descriptor on a particular Stream/Termination.

7.2 MG STUN client package

Package name: MG STUN Client

Package ID: mgstunc (0x00be)

Description: This package enables an MGC to determine the mapped IP address and port that will be routed back to the media component that sent the request. This package applies to both [IETF RFC 3489] and [IETF RFC 5389].

Version: 1

Extends: None

7.2.1 Properties

7.2.1.1 STUN address

Property name: STUN Address

Property ID: stuna (0x0001)

Description: This property indicates that the MG shall return a STUN mapped IP address and port for each local address indicated. The MGC may send this STUN mapped IP address and port to the peer as its remote IP address and port.

NOTE – Implementors should be aware that each STUN message to a STUN server may imply a 9.5-second delay (see section 9.1 of [IETF RFC 3489]). This may impact the time in which an ITU-T H.248 command reply can be sent to the MGC. In cases where the STUN delay is excessive the use of TransactionPending is encouraged.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element in the list of string SHALL be of type **`StunAddressReq`** in ABNF format:

```
StunAddressReq = "L"/("B"[COLON Stun-Transport-type])/
("S"[COLON Stun-Transport-type])
Stun-Transport-type = "UDP"/"TCP"/"TLS"
```

Where:

"L": Local Address – Do not perform STUN address mapping for the address in this position.

"B": Binding Request – Perform a STUN binding request on the address in this list position. STUN clients can communicate with a TURN server using UDP, TCP, or TLS over TCP.

"S": Shared Secret/Binding Request – Perform Shared Secret STUN messages before performing a STUN binding request on the address in this list position. The value of *STUN-TRANSPORT-TYPE* is "UDP", "TCP" or "TLS". These values are for "Binding Requests". "Shared Secret Requests" are always sent over TLS.

In an ITU-T H.248 command reply:

Each element of the string SHALL be of type **`StunAddressReply`** in ABNF format:

```
StunAddressReply = (IP4-address COLON PortNumber)/ (IP6-
address COLON PortNumber)/EToken [COLON ErrorCode]
PortNumber = UINT16
EToken = "E"
ErrorCode = 1*3(DIGIT); could be extended
```

The ABNF syntax of IP4-address and IP6-address is defined in [IETF RFC 4566].

The MG may reply with an IP4 or IP6 address, for example "192.168.1.10:10000" and "FF1E:03AD::7F2E:172A:1E24:20000",

or reply with an error information in which the Class and Number of the error (see section 11.2.9 of [IETF RFC 3489] or section 15.6 of [IETF RFC 5389]) is included.

Default: Empty List

Defined in: LocalControl

Characteristics: Read/Write

7.2.1.2 NAT lifetime

Property name: NAT Lifetime

Property ID: natl (0x0002)

Description: This property requests the MG to return the NAT binding lifetime associated with the transport address at a particular list position.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element may be one of the following characters:

T: "Time Request";

N: "No time request".

In an ITU-T H.248 command reply:

An integer (in string form) representing a Lifetime value,

Or:

E: "Error in list position", followed by the Class and Number of the error (section 11.2.9 of [IETF RFC 3489] or section 15.6 of [IETF RFC 5389]) is included in the response.

Default: Empty List

Defined in: LocalControl

Characteristics: Read/Write

7.2.1.3 RTO value

Property name: Retransmission timeout (RTO) Value

Description: This property requests the MG to set the initial retransmission timer RTO interval as defined in section 7.2.1 of [IETF RFC 5389].

Property ID: rto (0x0003)

Type: Integer

Possible values: 1 ms to 600000 ms (10 minutes)

Default: 100 ms

Defined in: LocalControl

Characteristics: Read/Write

7.2.2 Events

None.

7.2.3 Signals

None.

7.2.4 Statistics

None.

7.2.5 Error codes

None.

7.2.6 Procedures

To determine a STUN mapped address, the MGC shall issue an ITU-T H.248 Add/Modify/Move.req command containing the "*stuna*" property on the relevant termination/stream. If the MG wants to modify the value of RTO, the "*rto*" property should also be set on the relevant termination/stream. The MGC shall include a value in each position of list of string for each element described in the "*stunb/ac*" property.

For list positions designated with the value "*L*", the MG shall simply return an empty string for that position of the list.

For list positions designated with the value "*B*", the MG shall perform a STUN binding request messaging. If the binding request is successful, the MG shall return the mapped address in the list position of the reply. If the binding request is unsuccessful, then the STUN error shall be returned in the list position.

For list positions designated with the value "*S*", the MG shall perform a STUN Shared Secret request before issuing binding request messaging. The MG may perform a single Shared Secret request for all binding requests or issue multiple Shared Secret requests. STUN server implementations based on [IETF RFC 3489] do not support Shared Secret requests.

If the MGC wishes to reset the mappings, then it shall resend the parameter in an ITU-T H.248 request indicating "*L*", "*B*" or "*S*".

If the MGC wishes to receive the period of the binding lifetime with a NAT for an address it shall issue a "*natl*" property requesting "*T*" for each list element it wants a lifetime for and "*N*" for each element it does not want the time for. On receipt of the request, the MG will determine if a binding lifetime has been determined for a particular address. It will return the binding lifetime value if received from the NAT otherwise it will return "0". It is recommended that the "*natl*" property be included after the "*stuna*" property to maximize the possibility that a binding lifetime period is returned.

If the peer side returns error code 300 (Try Alternate), the transport address of the alternate server is in the ALTERNATE-SERVER attribute. The MG should use this transport address to resend the same request message to the alternate server. If the MG cannot send or the request still fails, the MG should indicate the error code to the MGC.

STUN server implementations based on [IETF RFC 3489] may not return a binding lifetime. The STUN information package should be used in this case.

7.3 MG TURN client package

Package name:	MG TURN Client
Package ID:	mgturnc (0x00bf)
Description:	This package enables an MGC to determine the mapped IP address and port used for data relaying through a TURN server. It enables the procedures defined by [IETF RFC 5766] to operate in a split MGC/MG environment.
Version:	1
Extends:	None

7.3.1 Properties

7.3.1.1 TURN address

Property name:	TURN Address
Property ID:	turna (0x0001)

Description: This property requests a particular local transport address and port that the MG performs a TURN allocate request on with a remote TURN server in order to reserve an address. As per section 6 of [IETF RFC 5766] the MGC may send this particular local transport address and port to the peer as its remote IP address and port.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element in the list of string SHALL be of type `TurnAddressReq` in ABNF format:

```
TurnAddressReq = ("A"
          [RequestedProps][TurnTransportType]
          [Lifetime][ChannelSend])/"N"
RequestedProps = SP "rp" COLON ReqP
TurnTransportType = SP "ttp" COLON TurnTranT
Lifetime = SP "l" COLON LifeT
ChannelSend = SP "c" COLON ChannelS
ReqP = 3* BinChar
BinChar = "0"/"1"
TurnTranT = "UDP"/"TCP"/"TLS"
LifeT = UINT32
ChannelS = "SEND"/"CHANNEL"
```

"A": Allocate Request – Perform a TURN Allocation request on the address in this list position. The REQUESTED-PROPS attribute may be indicated at the same time. The value of REQUESTED-PROPS is "000" to "111". The value of TURN-TRANSPORT-TYPE is "UDP" , "TCP" or "TLS". LIFETIME is the value of the LIFETIME attribute. The value of ChannelS is "SEND" or "CHANNEL". This flag is used to request the MG to send application data using ChannelData messages or using Send and Data indications.

"N": No change – Do not perform TURN address mapping for the address in this position.

In an ITU-T H.248 command reply:

Each element of the string contains a relay address, reflexive address and LIFETIME. Each element in the list of string SHALL be of type `TurnAddressReply` in ABNF format:

```
TurnAddressReply = ((IP4RelayAddr SP IP4ReflexiveAddr)
/ (IP6RelayAddr SP IP6ReflexiveAddr) SP LifeT ) / (EToken
[COLON ErrorCode])
IP4RelayAddr = IP4-address COLON PortNumber
IP4ReflexiveAddr = IP4-address COLON PortNumber
IP6RelayAddr = IP6-address COLON PortNumber
IP6ReflexiveAddr = IP6-address COLON PortNumber
PortNumber = UINT16
LifeT = UINT32
EToken = "E"
```

The ABNF syntax of IP4-address and IP6-address is defined in [IETF RFC 4566].

The MG may reply with the relay address, reflex address and the value of LIFETIME attribute, for example "192.168.1.10:10000 192.168.2.100:20000 11000" and "FF1E:03AD::7F2E:172A:1E24:20000 FF1E:03AD::7F2E:2AB3:9AA8:20000 30000". Or it may reply with an error indication in which the class and number of the error (see section 11.2.9 of [IETF RFC 3489] or section 15.6 of [IETF RFC 5389]) is included.

Default: Empty List

Defined in: LocalControl

Characteristics: Read/Write

7.3.1.2 TURN refresh

Property name: TURN Refresh

Property ID: turnr (0x0002)

Description: This property requests for a particular local transport address and port that the MG perform a TURN refresh request with a remote TURN server in order to keep the allocation and change the bandwidth or lifetime. As per section 7 of [IETF RFC 5766], a Refresh transaction can be used to either (a) refresh an existing allocation and update its time-to-expire, or (b) delete an existing allocation. MG should send periodic refresh request for each local transport address and port automatically. The MGC may request the MG to send a special refresh request to change bandwidth or lifetime via this property.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element in the list of string SHALL be of type **`RefreshReq`** in ABNF format:

```
RefreshReq = ("R"[COLON LifeT])/"N"
LifeT = UINT16
```

"R": Refresh Request – Perform a TURN refresh request on the address in this list position. Lifetime contains the value of the LIFETIME attribute. If the value of LIFETIME attribute is zero, this will cause the IP termination to remove the allocation, and all associated permissions and channel numbers. If a value for lifetime is not included a default lifetime as per [IETF RFC 5766] shall be assumed.

"N": No change – Do not perform TURN refresh request for the address in this position.

In an ITU-T H.248 command reply:

Each element of the string may be LIFETIME. Each element in the list of string SHALL be of type **`RefreshReply`** in ABNF format:

```
RefreshReply = LifeT/ EToken [:ErrorCode]
```

Lifetime contains the value of the LIFETIME attribute. As per section 7.2 of [IETF RFC 5766], in a successful response, the

LIFETIME attribute indicates the amount of additional time (the number of seconds after the response is received) that the allocation will live without being refreshed.

Or reply with an error information in which the Class and Number of the error (see section 11.2.9 of [IETF RFC 3489] or section 15.6 of [IETF RFC 5389]) is included.

Default: Empty List

Defined in: LocalControl

Characteristics: Read/Write

7.3.1.3 NAT lifetime

Property name: NAT Lifetime

Property ID: natl (0x0003)

Description: This property requests the MG to return the NAT binding lifetime associated with the transport address at a particular list position.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element may be one of the following characters:

T: "Time Request";

N: "No time request".

In an ITU-T H.248 command reply:

A string based integer representing a Lifetime value,

Or:

E: "Error in list position", followed by the class and number of the error (section 11.2.9 of [IETF RFC 3489]) is included in the response.

Default: Empty List

Defined in: LocalControl

Characteristics: Read/Write

7.3.2 Events

None.

7.3.3 Signals

None.

7.3.4 Statistics

None.

7.3.5 Error codes

None.

7.3.6 Procedures

To determine a TURN relayed address, the MGC shall issue an ITU-T H.248Add/Modify/Move.req command containing the "*turna*" property on the relevant termination/stream. The MGC shall include a value in each position of list of string for each element described in the "*stunb/ac*" property.

For list positions designated with the value "*N*", the MG shall simply return an empty string for that position of the list.

For list positions designated with the value "*A*", the MG shall perform a TURN allocate request messaging. If the allocate request is successful the MG shall return the relayed address, reflex address and the value of LIFETIME attribute in the list position of the reply. The following attributes may be derived from the "*turna*" property:

– EVEN-PORT, section 14.6 of [IETF RFC 5766];

– LIFETIME, section 14.2 of [IETF RFC 5766].

Communication between the TURN client and the TURN server can run over UDP, TCP or TLS. *"TurnTransportType"* in the syntax of the "*turna*" property is used to indicate the desired transport type.

When the TURN client has data to send to a peer, it may use either a ChannelData message or a Send indication. *"ChannelSend"* in the syntax of the "*turna*" property is used to indicate which method should be used.

If the allocate request is unsuccessful then the TURN error shall be returned in the list position.

If the MGC wishes to reset the mappings, then it shall resend the "*turna*" property in an ITU-T H.248 request indicating "*N*" or "*A*".

To refresh an existing allocation and update its time-to-expire, or to delete an existing allocation, the MGC shall issue a Modify/Move.req command containing the "*turnr*" property on the relevant termination/stream.

For list positions designated with the value "*R*" the MG shall perform TURN refresh request messaging. If the MGC wishes the TURN server to set the time-to-expire timer to something other than the default lifetime, the MGC indicates to the MG to include a LIFETIME attribute with the requested value. If the value of the LIFETIME attribute is 0, the TURN server immediately deletes the allocation. If the refresh request is successful, the MG shall return the LIFETIME attribute in the list position of the reply. If the refresh request is unsuccessful then the TURN error shall be returned in the list position.

7.4 MGC STUN client package

Package name: MGC STUN Client

Package ID: mgcstunc (0x00c0)

Description: This package enables an MGC to determine the STUN mapped IP address and port for a particular media component. This package allows the STUN client to remain at the MGC level rather than being implemented in the MG.

NOTE – This package has been included for backward compatibility reasons.

Version: 1

Extends: None

7.4.1 Properties

None.

7.4.2 Events

None.

7.4.3 Signals

7.4.3.1 MGC initiated STUN request

Signal name:	MGC Initiated STUN Request
Signal ID:	mgcistunr (0x0001)
Description:	The MGC shall send a signal for each media component address whose address is to be mapped. These signals may occur in the same ITU-T H.248 message. The STUN response messages are not used in this case. It is used to fulfil the case described in section 10.3 of [IETF RFC 3489].
Signal type:	Brief
Duration:	Not applicable

7.4.3.1.1 Binding request message

Parameter name:	Binding Request Message
Parameter ID:	brm (0x0001)
Description:	This parameter contains the MGC constructed STUN request message.
Type:	Octet String
Optional:	No
Possible values:	Any valid STUN request message.
Default:	None

7.4.3.1.2 Transport address

Parameter name:	Transport Address
Parameter ID:	ta (0x0002)
Description:	This parameter indicates the IP address of the media component that is to have its address STUN mapped.
Type:	Integer
Optional:	No
Possible values:	Any valid list position from the "stunb/ac" parameter.
Default:	None

7.4.4 Statistics

None.

7.4.5 Error codes

None.

7.4.6 Procedures

The MGC provides the STUN message contents, STUN server address and media address to the MG which will send a packet from the media address containing the STUN message contents to the STUN server address. The response will be provided to the MGC, thus any protocol or error handling can remain at the MGC level. This package enables the scenario described in section 10.3 of [IETF RFC 3489].

7.5 STUN information package

Package name: STUN Information

Package ID: stuni (0x00c1)

Description: This package enables an MGC to determine the type of NAT the MG is behind and the binding lifetime associated with the STUN server. Signals and events may be applied to the Root Termination.

Version: 1

Extends: None

7.5.1 Properties

None

7.5.2 Events

7.5.2.1 NAT type determination

Event name: NAT Type Determination

Event ID: nattd (0x0001)

Description: This event detects the end of the NAT Type etermination procedure (initiated by a NAT Determination signal) and returns the detected type.

7.5.2.1.1 EventsDescriptor parameters

None

7.5.2.1.2 ObservedEventsDescriptor parameters

7.5.2.1.2.1 NAT type

Parameter name: NAT Type

Parameter ID: natt (0x0001)

Description: The determined NAT type (scenario) as per section 10.1 of [IETF RFC 3489].

NOTE - The original STUN protocol [IETF RFC 3489] was obsoleted by [IETF RFC 5389]. The original NAT type discovery process as per section 10.1 of [IETF RFC 3489] was moved to a self-contained specification [IETF RFC 5780]. Section 4 of [IETF RFC 5780] describes the NAT type discovery process using NAT type naming according to [IETF RFC 4787]. [IETF RFC 6888] also uses the NAT type naming according to [IETF RFC 4787].

Type: Enumeration

Optional: No

Possible values: 0x0000: On the open Internet;

0x0001: Firewall that blocks UDP;

0x0002: Firewall that allows UDP out, and responses have to come back to the source of the request (like a symmetric NAT, but no translation). A symmetric UDP Firewall;

0x0003: Full-cone NAT;

0x0004: Symmetric NAT;

0x0005: Restricted cone NAT;

0x0006: Restricted port cone NAT.

Default: None

7.5.2.2 Binding lifetime determination

Event name: Binding Lifetime Determination

Event ID: bld (0x0002)

Description: This event detects the end of the Binding Lifetime determination procedure (initiated by a Binding Lifetime Determination signal) and returns the time period of the binding.

7.5.2.2.1 EventsDescriptor parameters

None.

7.5.2.2.2 ObservedEventsDescriptor parameters

7.5.2.2.2.1 Binding lifetime

Parameter name: Binding Lifetime

Parameter ID: bl (0x0001)

Description: The period of the life of the binding.

Type: Integer

Optional: No

Possible values: 0-3600 seconds

Default: None

7.5.3 Signals

7.5.3.1 NAT type determination

Signal name: NAT Type Determination

Signal ID: nattd (0x0001)

Description: This signal instructs the MG to detect the type of NAT that a particular media component IP address is behind.

Signal type: Brief

Duration: NA

7.5.3.1.1 Additional parameters

7.5.3.1.1.1 Transport address

Parameter name: Transport Address

Parameter ID: ta (0x0001)

Description: This parameter indicates the IP address of the media component that is to have its address STUN mapped.

Type: Integer

Optional: No

Possible values: Any valid list position from the "*stunb/ac*" parameter.

Default: None

7.5.3.2 Binding lifetime determination

Signal name: Binding Lifetime Determination

Signal ID: bld (0x0002)

Description: This signal instructs the MG to detect the binding lifetime with the NAT.

Signal type: Brief

Duration: NA

7.5.3.2.1 Additional parameters

7.5.3.2.1.1 Transport address

Parameter name: Transport Address

Parameter ID: ta (0x0001)

Description: This parameter indicates the IP address of the media component that is to have its address STUN mapped.

Type: Integer

Optional: No

Possible values: Any valid list position from the "*stunb/ac*" parameter.

Default: None

7.5.4 Statistics

None.

7.5.5 Error codes

None.

7.5.6 Procedures

7.5.6.1 NAT behaviour discovery as defined for original STUN protocol

This package is used to support the procedures as outlined in section 10 of [IETF RFC 3489].

7.5.6.2 NAT behaviour discovery as defined for updated STUN protocol

This package could be also used to support the procedures according to [IETF RFC 5780].

8 ICE support

The support of ICE implies that the SDP CANDIDATE attribute is supported. This can facilitate address gathering and description of candidates. However, due to the nature of the ITU-T H.248 connection model, several procedural changes are required.

The text below is structured based on the various ICE protocol phases, relevant SDP attributes and their SDP Offer/Answer representation in call control signalling like SIP. Thus, the reader should be familiar with the original ICE RFC [IETF RFC 5245].

Address gathering

The MGC may request an MG to perform address discovery through the use of the wildcarding mechanism on the CANDIDATE attribute. In this case the MG is responsible for STUN server discovery. Fully specifying the CANDIDATE attribute will not result in any action. The candidate-attribute is specified by the following ABNF [IETF RFC 5245]:

```
candidate-attribute = "candidate" ":" foundation SP component-id SP
                          transport SP
                          priority SP
                          connection-address SP     ;from [IETF RFC 4566]
                          port     ;port from [IETF RFC 4566]
                          SP cand-type
                          [SP rel-addr]
                          [SP rel-port]
                          *(SP extension-att-name SP
                            extension-att-value)
```

NOTE 1 – Section 4.5 of [IETF RFC 6544] defines an extended ABNF grammar for the SDP "a=candidate:" attribute for TCP.

NOTE 2 – *Trickle ICE* reuses above *vanilla ICE* syntax for the SDP "a=candidate:" attribute (see section 9.2 [b-IETF ice-trickle].

NOTE 3 – The priority value is the result of a particular address prioritization algorithm (see section 4.1.2.1 of [IETF RFC 5245] for basic rules and [b-IETF ice-dualstack] for guidelines related to ICE multihomed and IPv4/IPv6 dual stack fairness).

The MGC shall either include values for, or wildcard mandatory fields. Including values provides a way of narrowing down the selection. For example:

– If the MGC wanted a particular candidate type it could specify:

```
a=candidate:1 1 UDP  $ $ typ host
a=candidate:2 1 UDP  $ $ srflx raddr  $ rport $
```

– and the host address and server reflexive address would be returned.

– If the MGC wanted all candidate types it could specify:

```
a=candidate:$ $ $ $ $ $ typ *
```

– and the local, server reflexive address and relay addresses would be returned.

If a request results in multiple candidates, then these shall be returned in multiple "a=" lines.

As per section 4.3 of [IETF RFC 5245], if the MG utilizes RTCP, the MGC and MG MUST encode the RTCP candidate using the "a=rtcp" attribute as defined in [IETF RFC 3605]. If RTCP is not in use, the MGC and MG MUST signal that by using "b=RS:0" and "b=RR:0" as defined in [IETF RFC 3556].

If the MGC requires a MG to allocate a port for RTCP, the MGC will send the required information to request an RTCP port (e.g., "a=" line in SDP, a=rtcp $).

A further complication is provided by the selection of the "In-Use candidate" in the SDP media (m=) and connection (c=) lines. In the ITU-T H.248 Local Descriptor this is a local transport address in the MG. Whereas in ICE this may be another address. Therefore, in order to construct an SDP offer, the MGC shall determine the "In-Use candidate" from the CANDIDATE attributes returned in the response.

The *remote-candidate-att* is related to the SDP (SIP) offer and provides the identity of the remote candidates that the offerer wishes the answerer to use in its answers. It is assumed that the MGC (acting as the offerer) will populate this attribute based on candidate information from the MG.

Similarly, the ICE SDP "*ice-pwd-att*" attribute may be wildcarded or provided.

If the MGC asks for candidates, it should provide or ask for credentials in order for the MG to be able to perform the necessary signalling.

If MGC provides credentials, the MGC will send the required "a=" lines in SDP. For example:

```
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
```

If the MGC asks the MG to provide the credentials, the MGC will send the required "a=" lines in SDP. For example:

```
a=ice-pwd:$
a=ice-ufrag:$
```

If MGC does not supply the necessary "a=" lines in SDP, the MG may determine these credentials and return them to the MGC in a reply message. This is in order for the MG to be able to send a correct STUN message.

The use of the "*a=ice-lite*" attribute indicates whether or not the agent implements a lite version of ICE (see also clause 6.1.6 and Appendix III). As the agent resides in the MG this information needs to be provided to the MGC in order for the information to be communicated to a peer. Given the structure of the "*a=ice-lite*" attribute the MGC is unable to perform a CHOOSE wildcard operation on the attribute. As such where the MG implements a "lite" version of ICE it shall include the "a=ice-lite" attribute in any command response containing the candidate attribute.

Where a binary encoding of ITU-T H.248 is used, the ICE-related attributes may be realized through the use of [ITU-T H.248.1] Annex C.11 properties. In order to facilitate multiple values the "sub-list of" form should be used.

Connectivity checking

In order for an MG to perform some ICE operations, the remote candidate list is required. This remote candidate list can be provided to the MG in the form of candidate attributes being placed in the appropriate RDs. The MGC may be required to reformat the SDP answer/offer received (i.e., via SIP) in order for it to be applicable to ITU-T H.248 structures. The setting of the candidate attributes in the RD will not in itself generate any ICE operation such as Connectivity Checking or keep alive mechanisms. ITU-T H.248 packages are used to trigger these operations.

8.1 MG act-as STUN server package

Package name:	MG Act-as STUN Server
Package ID:	mgastuns (0x00c2)
Description:	This package enables an MGC to request that a particular address on an MG act as a STUN server in order to process, receive binding requests and return STUN binding responses. The purpose of the package is to enable the procedures defined in section 7.2 of [IETF RFC 5389].
Version:	1
Extends:	None

8.1.1 Properties

8.1.1.1 Act-as STUN server

Property name: Act-as STUN Server

Property ID: astuns (0x0001)

Description: This property requests for a particular local transport address that the MG be prepared to receive STUN binding requests for the purpose of connectivity checking.

Type: List of String

Possible values: In an ITU-T H.248 command request:

Each element in the list of string SHALL be of type **`MGActServer`** in ABNF format:

```
MGActServer = GroupID "|" Foundation "|" Component-id
"|" RequestType
GroupID = UINT16
Foundation = UINT16
Component-id = UINT16
RequestType = "N"/"S"
```

GroupId is as per UINT16 in clause B.2 of [ITU-T H.248.1].

Component-id is as per <component-id> in section 15.1 of [IETF RFC 5245].

Foundation is as per <foundation> in section 15.1 of [IETF RFC 5245].

RequestType is:

N ("No request"): do not process STUN requests;

S ("STUN Server"): act as a STUN server receiving binding requests.

Default: "STUN Server"

Defined in: LocalControl

Characteristics: Read/Write

8.1.2 Events

None.

8.1.3 Signals

None.

8.1.4 Statistics

None.

8.1.5 Error codes

None.

8.1.6 Procedures

To request the MG to act as a STUN server, the MGC should issue an ITU-T H.248 Add/Modify/Move.req command containing the "*astuns*" property on the relevant Termination/Stream.

8.2 Originate STUN continuity check package

Package name: Originate STUN Continuity Check

Package ID: ostuncc (0x00c3)

Description: This package enables an MGC to initiate a STUN continuity check binding request process at MG level. The purpose of the package is to enable the procedures defined in section 7 of [IETF RFC 5389].

Version 1 supported UDP, version 2 of this package indicates in addition the support of TCP.

Version: 2

Extends: None

8.2.1 Properties

8.2.1.1 Host candidate realm

Property name: Host Candidate Realm

Property ID: hcr (0x0001)

Description: This property indicates the realm the MG allocates IP address and port for host candidate. As per section 1 of [IETF RFC 5245], because ICE exchanges a multiplicity of IP addresses and ports for each media stream, it also allows for address selection for multi-homed and dual-stack hosts. As per section 4.1.1.1 of [IETF RFC 5245], host candidates are obtained by binding to ports (typically ephemeral) on an IP address attached to an interface (physical or virtual, including VPN interfaces) on the host. The information of host candidate is described in local SDP. If MGC requires the MG to gather more than one host candidate via local SDP, it uses this property to indicate to the MG to gather each of those host candidates in the indicated IP realm or VPN.

The property is applicable for UDP- and TCP-related ICE because the UDP- and TCP-specific syntax elements of the SDP "a=candidate:" atributes is not part of the property value syntax.

Type: List of String

Possible values: Each element in the list of string SHALL be of type **HostCandRealm** in ABNF format:

```
HostCandRealm = GroupID "|" Foundation "|" Component-id
"|" Realm
GroupID = UINT16
Foundation = UINT16
Component-id = UINT16
Realm = ALPHA 0*63(ALPHA/ DIGIT)
ALPHA = %x41-5A / %x61-7A ; A-Z / a-z
DIGIT = %x30-39; 0-9
```

Where:

GroupId is as per UINT16 in clause B.2 of [ITU-T H.248.1].

Component-id is as per <component-id> in section 15.1 of [IETF RFC 5245].

Foundation is as per <foundation> in section 15.1 of [IETF RFC 5245].

Realm is a string used to discriminate overlapping IP address spaces. It is the identifier of the IP domain or VPN.

e.g.,

```
"1|1|1|realm1", "1|2|1|realm2"
```

And MGC sends such a local SDP to a MG (here an ICE for UDP example):

```
v=0
c=IN IP4 $
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
m=audio $ RTP/AVP 0
a=candidate:1 1 UDP 2130706431 $ $ typ host
a=candidate:2 1 UDP 2113929215 $ $ typ host
```

The MG should gather two host candidates. The first one is in IP realm "realm1" and the second one is in IP realm "realm2".

Default: None

Defined in: LocalControl

Characteristics: Read/Write

8.2.2 Events

8.2.2.1 Connectivity check result

Event name: Connectivity Check Result

Event ID: ccr (0x0001)

Description: This event returns the result of an ICE connectivity check.

8.2.2.1.1 EventsDescriptor parameters

None.

8.2.2.1.2 ObservedEventsDescriptor parameters

8.2.2.1.2.1 Candidate/transport pair

Parameter name: Candidate/Transport Pair

Parameter ID: ctp (0x0001)

Description: The list of the successful candidate/transport pairs.

The Event parameter is applicable for UDP- and TCP-related ICE because the reported IP transport address value(s) is/are layer 4 (L4) protocol agnostic.

Type: List of String

Optional: No

Possible values: Each element in the list of string SHALL be of type `CandPair` in ABNF format:

```
CandPair = StreamID "|" GroupID "|" Foundation-l "|"
```

```
        Foundation-r "|" Component-id [Lp-connection-
        address] [Rp-connection-address] [Rc]
StreamID = UINT16
GroupID = UINT16
Foundation-l = UINT16
Foundation-r = UINT16
Component-id = UINT16
Lp-connection-address = "|lp-" Connection-address COLON
PortNumber
Rp-connection-address = "|rp-" Connection-address COLON
PortNumber
Connection-address = IP4-address
Rc = "|RC"
PortNumber = UINT16
```

The ABNF syntax of IP4-address is defined in [IETF RFC 4566].

Where:

StreamID is as per StreamId in clause B.2 of [ITU-T H.248.1].

GroupID is as per UINT16 in clause B.2 of [ITU-T H.248.1].

Foundation-l (Local Foundation) is as per <foundation> in section 15.1 of [IETF RFC 5245].

Foundation-r (Remote Foundation) is as per <foundation> in section 15.1 of [IETF RFC 5245].

Component-id is as per <component-id> in section 15.1 of [IETF RFC 5245].

Lp (Local) Peer reflexive candidate as described in section 7.1.3.2 of [IETF RFC 5245].

Rp Peer reflexive remote candidate as described in section 7.2.1.3 of [IETF RFC 5245].

Rc (Role Change) is a flag to indicate that the control role has changed (i.e., controlled or controlling). Control role is described in section 7.2.1.1 of [IETF RFC 5245].

e.g.,

```
"1|2|3|4|1|lp-202.2.3.4:1000"
```

Means that StreamID is 1, GroupID is 2, Local Foundation is 3, Remote Foundation is 4, Component-id is 1, and there is a local peer reflexive candidate in this candidate pair. The IP address of this local peer reflexive candidate is 202.2.3.4, and the port is 1000.

Default: None

8.2.2.2 New peer reflexive candidate

Event name: New Peer Reflexive Candidate

Event ID: nprc (0x0002)

Description: This event indicates that a new peer reflexive candidate was discovered during a connectivity check.

8.2.2.2.1 EventsDescriptor parameters

None.

8.2.2.2.2 ObservedEventsDescriptor parameters

8.2.2.2.2.1 Candidate

Parameter name: Candidate

Parameter ID: can (0x0001)

Description: A list of the newly discovered peer reflexive candidates.

The Event parameter is applicable for UDP- and TCP-related ICE because the reported IP transport address value(s) is/are L4 protocol agnostic.

Type: Sub-list of String

Optional: No

Possible values: Each element in the list of string SHALL be of type **peerCand** in ABNF format:

```
peerCand = StreamID "|" GroupID "|" Foundation "|"
          Component-id "|" Connection-address COLON
           PortNumber
StreamID = UINT16
GroupID = UINT16
Foundation = UINT16
Component-id = UINT16
Connection-address = IP4-address
PortNumber = UINT16
```

The ABNF syntax of IP4-address is defined in [IETF RFC 4566].

Where:

StreamID is as per StreamId in clause B.2 of [ITU-T H.248.1].

GroupID is as per UINT16 in clause B.2 of [ITU-T H.248.1].

Foundation is as per <foundation> in section 15.1 of [IETF RFC 5245].

Component-id is as per <component-id> in section 15.1 of [IETF RFC 5245].

A newly discovered peer reflexive candidate as described in section 7.1.3 of [IETF RFC 5245].

e.g.,

```
"1|2|3|4|202.2.3.4:1000"
```

Means that the StreamID is 1, GroupID is 2, Foundation is 3, Component-id is 4, and a new peer reflexive candidate was discovered. The IP address of this peer reflexive candidate is 202.2.3.4, and the port is 1000.

Default: None

8.2.3 Signals

8.2.3.1 Send connectivity check

Signal name: Send Connectivity Check

Signal ID: scc (0x0001)

Description:	This signal initiates connectivity checking procedures as defined in section 7 of [IETF RFC 5245].
Signal type:	Brief
Duration:	NA

8.2.3.1.1 Additional parameters

8.2.3.1.1.1 Control

Parameter name:	Control
Parameter ID:	cntrl (0x0001)
Description:	This parameter indicates the controlling role defined in section 7.1.2.2 of [IETF RFC 5245].
Type:	Enumeration
Optional:	Yes
Possible values:	Controlling (0x0001): MG acts as controlling role Controlled (0x0002): MG acts as controlled role
Default:	Controlling
Characteristics:	Read

8.2.3.2 Send additional connectivity check

Signal name:	Send Additional Connectivity Check
Signal ID:	sacc (0x0002)
Description:	This signal instructs a MG to initiate additional ICE connectivity check procedures in the case additional candidate/transport address pairs are identified. The MG shall use the CANDIDATE attributes in both the Local and Remote Descriptors to determine the candidate/transport address pairs to use for the connectivity check. The MG shall resume and only perform checks not already performed.
Signal type:	Brief
Duration:	NA

8.2.3.2.1 Additional parameters

8.2.3.2.1.1 Control

Parameter name:	Control
Parameter ID:	cntrl (0x0001)
Description:	This parameter indicates the controlling role defined in section 7.1.2.2 of [IETF RFC 5245].
Type:	Enumeration
Possible values:	Controlling (0x0001): MG acts as controlling role Controlled (0x0002): MG acts as controlled role
Default:	Controlling
Characteristics:	Read

8.2.4 Statistics

None.

8.2.5 Error codes

None.

8.2.6 Procedures

8.2.6.1 IP transport protocol indication

STUN connectivity check procedures for UDP (section 7 of [IETF RFC 5245]) and TCP (section 7 of [IETF RFC 6544]) are largely overlapping, but there are also some L4 protocol specific aspects. Hence, the MG must know the specific STUN-over-L4 protocol layering.

NOTE – The following related item is for further studies:

– When L4-specific STUN connectivity checks are executed in L4-specific ITU-T H.248 streams, then there is no semantic problem due to the explicit L4 protocol indication in the ITU-T H.248 Stream Descriptor (i.e., the LD and/or RD). However, the situation in ICE/STUN based NAT-T is rather the opposite: there is a 1:1 mapping of ALL STUN connectivity checks to either (see ITU-T H.248 stream grouping model according to [ITU-T H.248.97]):

a) a single ITU-T H.248 component stream; or

b) a single ITU-T H.248 (de-)aggregation stream.

– There might be a gap concerning the indication of the specific L4 protocol type in an appropriate ITU-T H.248 signalling element.

– Furthermore, there are protocol stack cases:

a) application specific layering such as "SCTP-over-DTLS-over-L4"; and

b) generic layering "X-over-L4" cases. Leading to the basic question whether the ITU-T H.248.50 NAT-T toolkit should become "L4+ aware" or remain purely "L4+ agnostic" (as currently the case)?

8.2.6.2 Connectivity check procedures for UDP

When the "Send Connectivity Check" (*scc*) Signal is sent, the MGC should initiate the STUN continuity check procedures as outlined in section 7 of [IETF RFC 5245]. As the ICE agent functionality is split between a MGC and MG and the MG is responsible for sending the connectivity checks, the agent role (controlled or controlling, see section 5.2 of [IETF RFC 5245]) should be provided. In order for the MGC to obtain the result of the connectivity check, it shall set the "Connectivity Check Result" (*ccr*) event.

The usage of the "Send Connectivity Check" Signal shall clear the results of any previous connectivity checking on the Termination/Stream.

If the *ccr* Event is set and the *scc* Signal is received, the MG shall apply the procedures of section 7.1.2 of [IETF RFC 5245] to form and prioritize a checklist.

NOTE 1 – The calculation of the priority value basically follows section 4.1.2.1 of [IETF RFC 5245], though, there might be other policies such as [b-IETF ice-dualstack].

Once the procedures of section 7.1.3 of [IETF RFC 5245] have been completed, the MG shall then notify the MGC of the result. The MG shall deem the procedures of section 5.8 of [IETF RFC 5245] completed once all checks have reached the state "succeeded" or "failed". In order to uniquely identify the checks on the termination, the MG shall provide the StreamID, GroupID, foundation of the local candidate, foundation of the remote candidate and the component-id. If the control role of this candidate pair changes, an optional flag "Rc" is included in the end. The MG shall maintain the results of the checking for the lifetime of the Termination/Stream or until they are cleared by a subsequent "Send Connectivity Check" signal.

Where there is only one stream or group the respective ids default to 1.

If there are no successful candidate pairs for one of the particular component-ids in the list, then the connectivity check for that component-id has failed.

If there are more than one successful candidate pairs for a particular component-id in the list, only the candidate pair having the highest priority will be reported. The "Sub-list of" shall be in the same order as the checklist in order to maintain relative priority.

Depending on its connection role, "controlled" or "controlling", the MGC may use the result in the ObservedEventsDescriptor parameter *ccr/ctp* to determine which transport pair to use for the media connection.

If a new peer reflexive candidate is discovered as per the procedure in section 7.1.3.2.1 of [IETF RFC 5245], and if the "New Peer Reflexive Candidate" (*nprc*) Event is set, the MG will notify the MGC with the new peer reflexive candidate. In order to identify this peer reflexive candidate, the MG shall provide the StreamID, GroupID, Foundation, Component-id and the transport IP address and port number of this peer reflexive candidate. If the MGC requires that the peer reflexive candidate be paired with other remote candidates besides the one in the valid pair that will be generated, the MGC may generate an updated offer which includes the peer reflexive candidate. This will cause it to be paired with all other remote candidates. The list of the candidate pairs is updated in this case.

When a new peer reflexive candidate is discovered, the Connectivity Checks procedure continues. If the MGC updates the local candidates via the SDP in the ITU-T H.248 message, the MG will continue with any connectivity checks that are still in progress. It will not perform checks on the new candidates. The MGC shall send the "Send Additional Connectivity Check" (*sacc*) Signal to perform checking of the new candidates.

In some kinds of conditions, where the list of the connectivity check candidate pairs is changed (i.e., a new stream is added, an existing stream is modified or a new peer reflexive candidate is discovered) additional connectivity check procedures may be initiated. The MGC may send the "Send Additional Connectivity Check" Signal to initiate a connectivity check procedure on any candidate/transport pair not previously checked. In this procedure, the MG shall only perform checks not already performed. Another example is where a new stream is added after the connectivity checks for the existing stream have finished. In this case, the additional connectivity checks will only check the candidate pairs of the newly added stream.

8.2.6.3 Connectivity check procedures for TCP

This package may also be applied for TCP; the procedures of previous clause 8.2.6.2 are applicable under consideration of TCP specific aspects according to section 7 of [IETF RFC 6544].

8.3 STUN consent freshness package

Package name:	STUN Consent Freshness
Package ID:	stnconfres (0x0120)
Description:	This package allows an MGC to request an MG to perform the STUN usage for Consent procedures described in [IETF RFC 7675]. This allows the MG to send STUN binding requests and receive STUN binding responses in order to determine whether application data may be sent on the Stream.
Version:	1
Extends:	None

8.3.1 Properties

None.

8.3.2 Events

8.3.2.1 Consent State

Event name: Consent State

Event ID: constate (0x0001)

Description: This event allows a MG to indicate to the MGC whether consent has been granted or not for a particular stream. It is only reported if the state changes between "granted" and "not granted" (i.e. revoked) states and vice versa. In consent "not granted" state application data stops flowing. In consent "granted" state application data flows.

8.3.2.1.1 EventsDescriptor parameters

8.3.2.1.1.1 Request States

Parameter name: Request States

Parameter ID: reqstate (0x0001)

Description: The parameter allows the MGC to request notification of a change to "granted" and/or "not granted" state.

Type: Enumeration

Optional: Yes

Possible values:

G	Notify a transition to state "granted".
N	Notify a transition to state "not granted".
B	Notify a transition to state "granted" or "not granted".

Default: B

8.3.2.1.2 ObservedEventsDescriptor parameters

8.3.2.1.2.1 State

Parameter name: State

Parameter ID: state (0x0001)

Description: This parameter indicates the consent state that the transport has transitioned to.

Type: Enumeration

Optional: No

Possible values:

G	Consent has been granted and application data is flowing.
N	Consent is not granted and application data is not flowing.

Default: None

8.3.2.2 STUN Consent Request Failure

Event name: STUN Consent Request Failure

Event ID: confail (0x0002)

Description:	This event allows a MG to indicate to the MGC when a response to a STUN binding request indicating the failure to renew consent.

8.3.2.2.1 EventsDescriptor parameters

None.

8.3.2.2.2 ObservedEventsDescriptor parameters

None.

8.3.3 Signals

8.3.3.1 Consent Test

Signal name:	Consent Test
Signal ID:	contst (0x0001)
Description:	This signal initiates the combined consent freshness test using STUN request/response procedures as outlined in section 5.1 of [IETF RFC 7675].
Signal Type:	On/Off
Duration:	Not applicable.

8.3.3.1.1 Additional parameters

8.3.3.1.1.1 Test Interval

Parameter name:	Test Interval
Parameter ID:	tstint (0x0001)
Description:	This parameter indicates the consent check interval as defined by section 5.1 of [IETF RFC 7675]. The MG sends consent checks at random intervals between 0.8N and 1.2N.
Type:	Integer
Optional:	Yes
Possible values:	4000 ms upwards NOTE – The typical test interval is at the time scale of seconds and tens of seconds. Small interval sizes are not recommended due to significant messaging and processing overhead that this would introduce.
Default:	5000 ms

8.3.4 Statistics

None.

8.3.5 Error codes

None.

8.3.6 Procedures

8.3.6.1 STUN Consent Freshness initiation

To initiate the STUN consent freshness procedures from [IETF RFC 7675] the MGC shall send the "Consent Test" (*stnconfres/contst*) Signal on the ITU-T H.248 stream where the consent is required. The *stnconfres/contst* Signal should be sent prior to or in the same ITU-T H.248 command as any signals initiating ICE connectivity checking to ensure that consent is granted as a result of successful initial ICE procedures.

On reception of the *stnconfres/contst* Signal the MG shall initiate the consent procedures in section 5.1 of [IETF RFC 7675]. As described the MG must not send application data (e.g., RTP, RTCP, SCTP, datagram transport layer security (DTLS)), over any transport protocol (e.g., UDP, TCP) on an ICE-initiated connection unless the receiving endpoint consents to receive the data. After a successful ICE connectivity check on a particular transport address, subsequent consent shall be obtained following the procedures described [IETF RFC 7675]. The MG shall send a STUN binding request consent check at the random interval indicated by the "Test Interval" (*tstint*) parameter in order to maintain consent.

When consent is granted the MG may send application data on the Stream. The granting of consent does not override other methods for controlling the flow of application or transport related data such as the Local Control StreamMode property [ITU-T H.248.1].

In order to be notified about the consent state the MGC should set the "Consent State" (*stnconfres/contst*) Event with the required states on the applicable Stream. As the Event indicates state transitions, the MGC should set the Event when sending the *stnconfres/contst* Signal to ensure the initial transition from consent "not-granted" (i.e. the default state before the completion of ICE connectivity check when the *stnconfres/contst* Signal is active) to a granted state after initial consent checking is reported.

The state transitions to a consent "granted" state on receipt of a valid STUN binding response. The state transitions to a consent "not-granted" state when a valid STUN binding response corresponding to one of the STUN requests sent in the last 30 seconds has not been received from the remote peer's Transport Address or the MG detects an immediate revocation of consent via one of the methods in section 4.2 of [IETF RFC 7675]. A MGC may also be notified of the revocation of consent based on the detection of Events related to a closure of the transport connection, i.e., a TLS *BNCChange* Event indicating "bearer release" [ITU-T H.248.90].

8.3.6.2 STUN consent request failure

In order to be notified of a failed STUN consent refresh the MGC shall set the "STUN Consent Request Failure" (*stnconfres/confail)* Event on the applicable Stream. When the MG detects a failure of a STUN consent refresh to renew the consent it shall send a Notify.Req with the *stnconfres/confail* ObservedEvent. The MGC may then decide to continue to allow the MG to send STUN consent refreshes, or stop STUN consent refreshes or to take another action as appropriate.

8.3.6.3 STUN Consent Freshness deactivation

To stop sending STUN consent requests the MGC should turn the *stnconfres/contst* Signal "off" by removing it from the applicable stream via a Modify.req command. On reception of the Signal the MG shall stop sending STUN consent refreshes. If any of the *stnconfres* Package Events are active on the Stream, then the MG shall stop sending application data at the expiry of the consent period. If no *stnconfres* Package Signals or Events are active on the Stream, then the STUN consent refresh procedures are no longer applicable to the Stream and application data may flow.

8.3.6.4 STUN Consent Freshness response

In order for a MG to respond to STUN binding requests used for consent refresh the MGC shall use the "MG act-as STUN server" package (clause 8.1).

8.3.6.5 Connection liveness

If the STUN consent freshness mechanism is used, no additional keepalives are needed to maintain liveness. E.g., such as those initiated through the use of the "Keep alive request" package (see clause 9.2).

Keepalives can be initiated through the use of the "Keep alive request" package (clause 9.2).

9 Keep-alive and pinhole support

The packages in this clause provide a means of opening a pinhole through a NAT and to maintain a binding with the NAT. It does not rely on ICE techniques.

9.1 MGC-originated STUN request package

Package name: MGC Originated STUN Request

Package ID: mgcostunr (0x00c4)

NOTE 1 – Section 10 of [IETF RFC 5245] makes use of this mechanism.

Description: The MGC may also periodically request the MG to send a STUN request in order to keep the binding with the NAT "alive". This should be done before the expiry of the keep-alive period.

NOTE 2 – This package is an exception to the rule in the scope that the MG determines the address to which a Binding Request is sent.

NOTE 3 – The package deals with IP address information, which is direction specific (source and destination) from an IP host perspective. The package description qualifies the source and destination specific parts, however, there is basically the assumption of address symmetry (i.e., the *local source address* is equal to the *local destination address*, and the same for remote addresses).

Version: 1

Extends: None

9.1.1 Properties

None

9.1.2 Events

9.1.2.1 STUN binding request failure

Event name: STUN Binding Request Failure

Event ID: fail (0x0001)

Description: This event is triggered if a STUN binding request has failed.

9.1.2.1.1 EventsDescriptor parameters

9.1.2.1.1.1 From address

Parameter name: From Address

Parameter ID: fa (0x0001)

Type: List of String

Optional: No

Possible values: "Y" (Yes): Failure reporting is required for this list position.
"N" (No): Failure reporting is not required for this list position.

9.1.2.1.2 ObservedEventsDescriptor parameters

9.1.2.1.2.1 Failure

Parameter name: Failure

Parameter ID: fail (0x0001)

Description: This parameter contains the reason for the failure.

Type: String

Optional: No

Possible values: If the response of STUN binding request is "time out", error code 1000 follows the <list position>.

If the mapping address in the response of STUN binding request changes, error code 1001 and the new mapping address follows <list position>. The format is "<list position>:IP address ":" port. e.g., "1:202.1.2.3:1000"

Default: None

9.1.3 Signals

9.1.3.1 Send STUN request

Signal name: Send STUN Request

Signal ID: sstunr (0x0001)

Description: This signal instructs a MG to send a binding request from the local source address to the remote destination address as contained in the RD.

Signal type: Brief

Duration: NA

9.1.3.1.1 From address

Parameter name: From Address

Parameter ID: fa (0x0001)

Description: This parameter contains the local source addresses where the binding request should be sent from, to the corresponding remote destination address.

Type: List of String

Optional: No

Possible values: *N* ("No Request"): do not perform a STUN binding request from the local source address.

Default: None

9.1.3.1.2 Retransmission time interval

Parameter name: Retransmission Time Interval

Parameter ID: rti (0x0002)

Description: This parameter contains the initial retransmission time interval for the STUN request, as RTO defined in section 7.2.1 of [IETF RFC 5389].

Type: Integer

Optional: Yes

Possible values:	1 ms to 600000 ms (10 minutes)
Default:	100 ms

9.1.4 Statistics

None.

9.1.5 Error codes

None.

9.1.6 Procedures

The MGC should trigger the MG to initiate a STUN binding request to the remote address when it suspects that the local address is behind a restricted cone NAT. This will have the effect of opening a pinhole, allowing the remote end to send packets to the local end. If the MGC initiates a STUN binding request to the remote address, it can keep NAT bindings active. If there are no packets sent between the local and remote address pairs being used for media for *Tr* seconds (where packets include media and previous keep-alives; variable *Tr* refers to Signal parameter value *mgcostunr/sstunr/rti*), the MG MUST generate a keep-alive on that pair. To detect STUN binding request failures the *fail* Event should be set.

This procedure is distinct from the ICE -related STUN continuity check procedures.

9.2 Keep alive request package

Package name:	Keep Alive Request
Package ID:	kar (0x00c5)
Description:	This package enables the MGC to request the MG to send a packet in order to open a pinhole through a server or to maintain a NAT binding. This packet goes through the same way as the media flows. For example it allows the techniques as defined in section 10 of [IETF RFC 5245]. NOTE – This package is an exception to the rule in the scope that the MG determines the address to which a Binding Request is sent.
Version:	1
Extends:	None

9.2.1 Properties

None.

9.2.2 Events

None

9.2.3 Signals

9.2.3.1 Send keepalive packet

Signal name:	Send Keepalive Packet
Signal ID:	skap (0x0001)
Description:	This signal instructs a MG to send a keepalive packet from the local source address to the remote destination address contained in the RD.
Signal type:	Brief

Duration: NA

9.2.3.1.1 From address

Parameter name: From Address

Parameter ID: fa (0x0001)

Description: This parameter contains the local source addresses where the keepalive packet should be sent from, to the corresponding remote destination address. The MGC shall include a value in each position of the list of string for each element as described in the "*stunb/ac*" property (see clause 7.1.1.1).

Type: List of String

Optional: No

Possible values: "Send" (*S*): send a keepalive packet from the local address.

"Not Send" (*N*): do not send a keepalive packet from the local address.

Default: None

9.2.3.1.2 Keep alive transmission interval

Parameter name: Keep Alive Transmission Interval

Parameter ID: ti (0x0002)

Description: This parameter contains the transmission time interval for sending the keepalive packet. The value of this parameter corresponds to the Tr timer from [IETF RFC 5245].

Type: Integer

Optional: Yes

Possible values: 15000 ms or more

Default: 15000 ms (15 seconds), unless provisioned otherwise.

9.2.3.1.3 Keep alive packet type

Parameter name: Keep Alive Packet Type

Parameter ID: kapt (0x0003)

Description: This parameter indicates the type of keep alive packet type.

Type: Enumeration

Optional: Yes

Possible values: et (0x0000): Transport (i.e., UDP, datagram congestion control protocol (DCCP)) packet of 0-byte;

rm (0x0001): RTCP packets multiplexed with RTP packets;

sbi (0x0002): STUN binding indication;

cn (0x0003): RTP packet with comfort noise payload;

no (0x0004): Reserved for RTP packet with No-Op payload (see [b-IETF no-op]);

iv (0x0005): RTP packet with incorrect version number;

up (0x0006): RTP packet with unknown payload type.

NOTE – Value no (0x0004) is reserved for a future payload type.

Default: up (0x0006): RTP packet with unknown payload type

9.2.4 Statistics

None.

9.2.5 Error codes

None.

9.2.6 Procedures

9.2.6.1 MGC stimuli for IP bearer path keep-alive requests

The IP bearer path related keep-alive functions are part of overall NAT traversal solutions. The MGC may trigger keep-alive procedures due to:

– local policy (provisioned policy, e.g., in case that all calls of an IP network domain require keep-alive support); or/and

– call-individual requests (and parameterization) by application control protocol signalling (see e.g., [b-IETF RFC 6223], [b-IETF RFC 5626], [b-IETF rkeep]).

Example: A SIP network with usage of [b-IETF RFC 6223]: the MGC could (inter alia) derive the Signal parameters "Keep Alive Transmission Interval" (*kar/skap/ti*) and "Keep Alive Packet Type" (*kar/skap/kapt*) from SIP signalling.

9.2.6.2 Triggering keep-alive at MG by MGC

In order to open or maintain a NAT binding, the MGC should send the "Send Keepalive" (*skap*) signal to the MG to initiate sending a keep-alive packet to the remote address. This will have the effect of opening a pinhole (or keeping a NAT binding open) allowing the remote end to send packets to the local end. The "Keep Alive Transmission Interval" (*ti*) parameter is used in order for the MG to autonomously send a keep-alive packet when no packets have been detected for the address for time interval Tr.

Where a MGC utilizes a "Send Keepalive Packet" request signal with the "RTP packet with unknown payload type", it may be required to communicate this to a peer MGC via the call contrrol signalling embedded SDP. An MGC on reception of this SDP attribute should then set this SDP attribute in the Remote Descriptor of the applicable MG Termination/Stream. It indicates to the MG that the remote end will utilize a keep-alive using an RTP packet with an unknown payload type.

10 Package-independent NAT-T procedures

10.1 Support for MG terminated STUN-based connectivity checks

10.1.1 Overview

STUN-based connectivity checks can be originated or terminated by the MG. Clause 8.2 defines support for MG originated STUN connectivity checks. This clause covers network scenarios where the MG is requested to terminate the STUN-based connectivity check procedure of STUN application ICE (e.g., model according Figure 2). Appendix I.1 illustrates an example use case in more detail.

NOTE – A MG which only supports termination of STUN-based connectivity check procedures necessarily only supports ICE-lite.

10.1.2 Required MG support functions

The MG is required to support:

– the STUN protocol [IETF RFC 5389] at the IP bearer interface of the MG (clause 10.1.3);

– the signalled credentials (for the bearer plane STUN-based connectivity checks) in ITU-T H.248 (clause 10.1.4);

– the ICE conclusion process (including media latching) (clause 10.1.5); and optionally

– reporting of latched addresses to the MGC (clause 10.1.5).

10.1.3 Indication of bearer type "STUN/L4/IP"

The MGC may indicate a STUN-enabled termination/stream endpoint by ICE-specific SDP in the LD/RD (see next clause).

10.1.4 Signalling of credentials

10.1.4.1 General behaviour

The motivation for credentials is described in sections 2.5 and 7 in [IETF RFC 5245], e.g.,

– *Each STUN connectivity check is covered by a message authentication code (MAC) computed using a key exchanged in the signaling channel. This MAC provides message integrity and data origin authentication, thus stopping an attacker from forging or modifying connectivity check messages*.

The two ICE-specific SDP attributes "ice-pwd" and "ice-ufrag" are used for that purpose. The MGC shall signal these SDP elements to the MG.

NOTE – See also section 7.1.2.3 of [IETF RFC 5245], which specifies that STUN's "short-term credentials" need to be used by ICE, and section 10.1 of [IETF RFC 5245], which defines this short-term credential mechanism, and which explains why username and password need to be exchanged via the signalling channel.

10.1.4.2 Detailed ITU-T H.248 signalling of credentials

The usage of credentials in the ITU-T H.248 Local Descriptor and Remote Descriptor, in the direction from MGC to MG (command request) and vice versa (command reply), depends on the applied ICE protocol variant, and whether the resource management of ICE credential is provided by the MGC or MG, is summarized in Table 2:

Table 2 – Usage of SDP-defined credentials in ITU-T H.248 LD and RD

No.	Resourcemanagement	ICE variant	SDP attributes (credentials)	MGC-to-MG (request)		MG-to-MGC (reply)	
				LD	RD	LD	RD
1.1	MG (Note1)	ICE-lite	"a=ice-ufrag:" "a=ice-pwd:"	Yes Yes	No No	Yes Yes	No No
1.2		full ICE	"a=ice-ufrag:" "a=ice-pwd:"	Yes Yes	Yes Yes	Yes Yes	No No
2.1	MGC (Note 2)	ICE-lite	"a=ice-ufrag:" "a=ice-pwd:"	Yes Yes	No No	O O	No No
2.2		full ICE	"a=ice-ufrag:" "a=ice-pwd:"	Yes Yes	Yes Yes	O O	No No

NOTE 1 – SDP would be wildcarded CHOOSE in case of MG level resource management.

NOTE 2 – To the optional (O) tag in the MG reply column: The MGC chooses ICE *ufrag* and *passwd* values and sends them to the MG in the requested Local Descriptor. If the MGC locally stores these values, then the MG is not required to reply them back to the MGC. Thus it may be optional for the MG to reply *ice-ufrag* and *ice-pwd* values to MGC.

10.1.5 ICE conclusion process (STUN-specific media latching)

The STUN connectivity check procedure provides an explicit indication to the MG about the existence of a remote NAT device in the IP bearer path, leading to the conclusion to perform latching.

NOTE – Background: this requirement relates to the *ICE conclusion process* as described in section 8 of [IETF RFC 5245]. This conclusion process depends on the supported ICE variant and, in case of full-ICE, on the ICE role (controlling or controlled).

The MG shall autonomously perform the ICE conclusion process which implies such STUN-specific latching, i.e., without any explicit indication by the MGC (as in case of [ITU-T H.248.37] stimulated latching). See also example in Appendix I.1.

The MGC may subscribe for optional address reporting (according to clause 7 of [ITU-T H.248.37]).

Appendix I

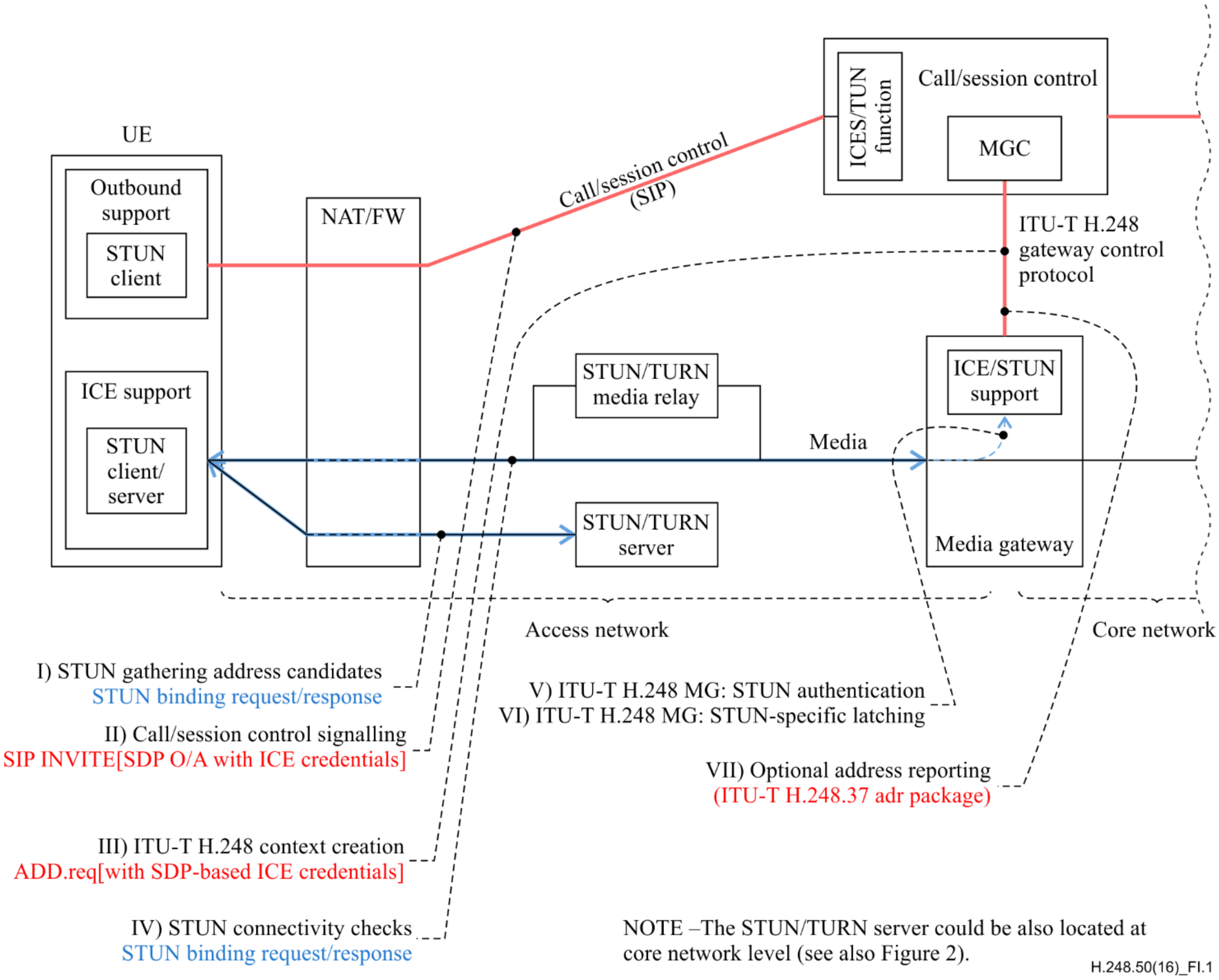
Example signalling scenarios for package-independent NAT-T procedures

(This appendix does not form an integral part of this Recommendation)

I.1 Example #1: ICE/STUN support by ITU-T H.248 IP access gateways

I.1.1 Network model

Figure I.1 illustrates a "half call" model, which is consistent with the network model of clause 6.1.1. An ITU-T H.248 IP-IP gateway is located between access and core network level. It's a widespread model for many IP-based network access technologies.



UE
Outbound support
STUN client
ICE support
STUN client/ server
NAT/FW
Call/session control (SIP)
ICES/TUN function
Call/session control
MGC
ITU-T H.248 gateway control protocol
STUN/TURN media relay
ICE/STUN support
Media
Media gateway
STUN/TURN server
Access network
Core network
I) STUN gathering address candidates
STUN binding request/response
II) Call/session control signalling
SIP INVITE[SDP O/A with ICE credentials]
III) ITU-T H.248 context creation
ADD.req[with SDP-based ICE credentials]
IV) STUN connectivity checks
STUN binding request/response
V) ITU-T H.248 MG: STUN authentication
VI) ITU-T H.248 MG: STUN-specific latching
VII) Optional address reporting
(ITU-T H.248.37 adr package)
NOTE –The STUN/TURN server could be also located at core network level (see also Figure 2).
H.248.50(16)_FI.1

Figure I.1 – Network model for ITU-T H.248 IP access gateways (derived from Figure 2 with scope on half call only)

The user equipment (UE) provides ICE-controlled STUN/TURN procedures according to clause 6.5.3.

I.1.2 Example signalling at SIP and ITU-T H.248 interfaces

Figure I.1 highlights the most important phases during the establishment of an outgoing call. The communication between the UE and STUN/TURN server as well as the UE and SIP servers is out of scope of this appendix. Only some significant SIP/SDP is indicated here.

The UE starts to gather addresses by contacting a STUN server (phase I; the UE receives the outgoing address of the NAT/FW (so called *server reflexive address*) by the STUN binding request/response cycle). After the *address prioritization* step (by the UE), an outgoing SIP message inclusive of an SDP offer is sent (phase II) with the purpose of advertisement of the candidate address(es). Table I.1 indicates the significant SDP for ICE.

Table I.1 – Example command encoding– (SIP) SDP Offer

(SIP) SDP encoding	Comments

v=0
o=...
s=...
t=...
m=audio <UE_A_port_audio_rtp> RTP/AVP …
c=IN IP6 <UE_A_IP_addr_audio_rtp>
…
a=candidate:...
a=ice-ufrag:<UE_A_ice_ufrag_audio>
a=ice-pwd:<UE_A_ice_pwd_audio>
…
	Two type of ICE-related SDP information:
	1. Address advertisement ("candidate(s)")
	2. Authentication and message-integrity mechanisms for STUN

The IP bearer path is routed through the MG. The subsequent STUN based connectivity checks will be therefore processed by the MG. The MGC prepares the MG corresspondingly (phase III) by creating a Context and stream endpoint (SEP). The SEP requires the information for STUN authentication, see Table I.2.

Table I.2 – Example command encoding – MGC request

ITU-T H.248 encoding (shortened command)	Comments

MGC to MG:
MEGACO/3 [11.9.19.65]:55555
Transaction = 1 {
  Context = $ {
    Add = ip/$/$/$ {
      Media {
        Stream = 1 {
          LocalControl {
            ipdc/realm = <access realm>,
            Mode = Inactive,
            ...
          },
          Local {
            v=0
            c=IN IP6 $
            m=audio $ RTP/AVP -
            a=ice-ufrag:<IP_A_ice_ufrag_audio> ; (Note 1)
            a=ice-pwd:<IP_A_ice_pwd_audio>   ; (Note 1)
          },
          Remote {
            v=0
            c=IN IP6 <UE_A_IP_addr_audio_rtp>
            m=audio <UE_A_port_audio_rtp> RTP/AVP -
            a=ice-ufrag:<UE_A_ice_ufrag_audio>
            a=ice-pwd:<UE_A_ice_pwd_audio>
          }
…
}
	NOTE 1 – Or wildcard CHOOSE in case of MG level management of resources "ICE username fragment" and "ICE password" (see also Table 2 in clause 10.1.4.2):
	a=ice-ufrag: $
	a=ice-pwd:$

A possible example MG reply is indicated in Table I.3.

Table I.3 – Example command encoding– MG reply

ITU-T H.248 encoding (shortened command)	Comments

MG to MGC:
MEGACO/3 [125.125.125.111]:55555
Reply = 1 {
  Context = <C1> {
    Add = <IP_A> {
      Media {
        Stream = 1 {
          Local {
            v=0
            c=IN IP6 <IP_A_IP_addr_audio>
            m=audio <IP_A_port_audio_rtp> RTP/AVP -
            a=ice-ufrag:<IP_A_ice_ufrag_audio>
            ; ice-pwd attribute might not be returned as MG does not perform STUN binding request authentication initially (Note 1)
…
          },
          Remote {
            v=0
            c=IN IP6 <UE_A_IP_addr>
            m=audio <UE_A_port_audio_rtp> RTP/AVP -
            a=ice-ufrag:<UE_A_ice_ufrag_audio>
            a=ice-pwd:<UE_A_ice_pwd_audio>
        },
	NOTE 1 – The "a=ice-pwd" value in this SEP's Local Descriptor is used by the UE when generating STUN binding requests (section 7.1.2.3 of [IETF RFC 5245], e.g., with agent "L = UE" and with agent "R = MG"). This SEP then, when receiving such STUN binding requests (thus as STUN server) needs to use it's own password (in the Local Descriptor) to check the received STUN binding request's validity (section 7.2 of [IETF RFC 5245] and section 10.1.2 of [IETF RFC 5389]).
	Therefore, the "a=ice-pwd" attribute in the Local Descriptor is actually needed. But, if the MG is an ICE-lite endpoint, it may ignore the" a=ice-pwd" attribute in the RD, because in this case it never generates and sends STUN binding requests itself (see also Table 2 in clause 10.1.4.2).

The MG monitors the IP bearer path and terminates STUN connectivity checks (phase V). The learnt addresses are used for latching, i.e., the adaption of the remote destination address (phase VI).

This kind of latching (as part of the "ICE conclusion process") is called (here) STUN-specific latching, similar to ITU-T H.248.37-stimulated latching. Both belong conceptionally to the overall function of "media latching" (as a kind of NAT traversal mechanism), but differ from ITU-T H.248 perspective.

NOTE – The difference:

a) "ITU-T H.248.37 latching": explicitly triggered latching via an ITU-T H.248 Signal;

b) "STUN-specific latching" (i.e., MG autonomously performs the ICE conclusion process (which incorporates latching of the peer's transport address, chosen by the controlling side)): implicitly triggered latching via "ICE/STUN" specific SDP.

The MGC could enable the address reporting capability (according to [ITU-T H.248.37]), which is normally not required for the progress of the call establishment process.

Appendix II

ICE for TCP

(This appendix does not form an integral part of this Recommendation.)

II.1 Introduction

The purpose of this appendix is to highlight major differences of *ICE for TCP* vs the original *ICE for UDP*. Only aspects relevant for ITU-T H.248 entities are indicated. Communication service specific aspects of ICE are out of scope of this Recommendation.

NOTE – For instance, the *Web Real-Time Communication* (WebRTC) service imposes a given protocol stack layering {DTLS|SRTP-over-UDP|TCP-over-IP} (see [b-IETF rtcweb-transports]), which leads e.g., to the layering of a single DTLS connection over UDP or TCP. Hence, there is a tight coupling between DTLS usage and ICE/STUN for UDP and TCP.

II.2 MG bearer interface: protocol stack

The "ICE for TCP" protocol stack (Figure II.1) is defined in section 3 of [IETF RFC 6544]:

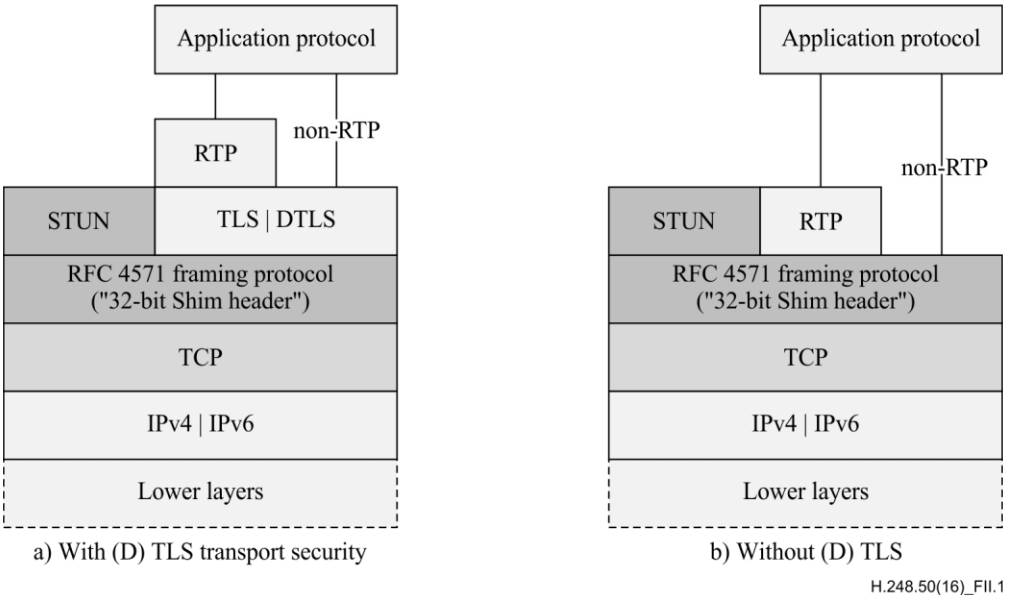


Figure II.1 – ICE for TCP – protocol stack (MG bearer interface)

The framing protocol (according to [b-IETF RFC 4571]) on top of TCP is mandatory. The transport level security protocols are optional. The option of using an UDP tunnel for TCP (section 5.4 of [IETF RFC 6544], "UDP-Tunneled Candidates") is not indicated in Figure II.1.

II.2 Bearer establishment

The STUN procedures (address gathering, connectivity checks, etc.) are executed over TCP bearer connection(s). The MG-side TCP-enabled ITU-T H.248 stream endpoints are controlled according to [b-ITU-T H.248.84] or/and [b-ITU-T H.248.89]. Additional (D)TLS transport level security may require support of [ITU-T H.248.90] or [b-ITU-T H.248.93], dependent on how far the MG is involved in (D)TLS traffic processing.

II.3 TCP address candidates

The UDP-candidate types [IETF RFC 5245],

```
candidate-types = "host" / "srflx" / "prflx" / "relay" / token
```

are further qualified by an extension type in case of ICE for TCP [IETF RFC 6244]:

```
tcp-type = "active" / "passive" / "so"
```

which is important concerning the direction of TCP bearer establishment (from MG perspective). Table II.1 summarizes the semantic of the TCP extension types:

Table II.1 – Type extensions for TCP-based candidates

Type extension for candidates	Semantic
active	An active candidate is one for which the agent will attempt to open an outbound connection but will not receive incoming connection requests.
passive	A passive candidate is one for which the agent will receive incoming connection attempts but not attempt a connection.
simultaneous-open (S-O)	An S-O candidate is one for which the agent will attempt to open a connection simultaneously with its peer.

This leads to the following TCP candidate type variations (Table II.2):

Table II.2 – TCP candidate options

Main candidate types	Variations
host	{active|passive|simultaneous-open} host candidate.
server reflexive	{active|passive|simultaneous-open} server reflexive candidate.
peer reflexive	{active|passive|simultaneous-open} peer reflexive candidate.
relayed	{active|passive|simultaneous-open} relayed candidate.
NAT-assisted	SDP encoded as "server reflexive", but with higher priority value.
UDP-tunneled	The "TCP(/IP) packet over UDP" tunnel leads to "ICE for UDP". SDP encoded as "relayed" or "server reflexive".

Example SDP grammar (ICE for TCP):

```
a=candidate:<> <> TCP <> <> typ host tcptype active
a=candidate:<> <> TCP <> <> typ host tcptype passive
a=candidate:<> <> TCP <> <> typ host tcptype so
a=candidate:<> <> TCP <> <> typ srflx raddr <> rport <> tcptype active
a=candidate:<> <> TCP <> <> typ srflx raddr <> rport <> tcptype passive
a=candidate:<> <> TCP <> <> typ srflx raddr <> rport <> tcptype so
a=candidate:<> <> TCP <> <> typ prflx raddr <> rport <> tcptype active
a=candidate:<> <> TCP <> <> typ prflx raddr <> rport <> tcptype passive
a=candidate:<> <> TCP <> <> typ prflx raddr <> rport <> tcptype so
a=candidate:<> <> TCP <> <> typ relay raddr <> rport <> tcptype active
a=candidate:<> <> TCP <> <> typ relay raddr <> rport <> tcptype passive
a=candidate:<> <> TCP <> <> typ relay raddr <> rport <> tcptype so
```

Appendix III

ICE mode support by gateway-embedded ICE agents

(This appendix does not form an integral part of this Recommendation.)

III.1 Background

The differences between ICE modes have to be extracted from [IETF RFC 5245] due to lack of an explicit description for an "ICE lite" capability set. Complementary ICE documentation provides only a very high level description, such as the following expired IETF draft [b-IETF ice-lite]:

```
"A Lite Ice Implementation (LII) behaves much like a normal ICE
implementation, with three major differences:
o It only gathers candidate addresses from its own interfaces.
o It cannot be a controlling endpoint.
o It does not generate checks but only responds to periodic checks
   from other endpoints."
```

III.2 Main differences between ICE-full and ICE-lite

The following Table III.1 provides an attempt to indicate the differences at a lower level, applicable for ICE implementations by ITU-T H.248 entities. Whenever there is a discrepancy between the information in this appendix and [IETF RFC 5245], the IETF RFC takes precedence. It should be noted that this Table provides some abstraction. Further differences could be identified when looking at a more detailed protocol level.

Table III.1 – High-level comparison of ICE modes

	Capability	Full mode	Lite mode	Comment
1	**Gathering Candidates:**			
1.1	Host candidates	Yes	Yes (Note 1)	"no real candidate gathering (because STUN/TURN not required)"
1.2	Server reflexive candidates	Yes	No	STUN/TURN based
1.3	Relayed candidates	Yes	No	TURN based
1.4	Peer reflexive candidates	Yes	No	Possible result of connectivity checks
1.5	**TCP-specific candidates:**			[IETF RFC 6544]
1.5.1	NAT-assisted candidates (TCP)	Yes	No	[IETF RFC 6544]
1.5.2	UDP-tunneled candidates (TCP)	Yes	No	[IETF RFC 6544]
2	**Prioritizing Candidates**			
2.1	Priority assignment	Yes	Yes	
2.2	Robust IP address selection (V4 / V6)	Yes (Note 2)	No	
3	**SIP: generate SDP Offer (or SDP Answer)**			
3.1	Choosing default candidates	Yes	Yes	
3.2	Insert SDP attribute "a=ice-lite"	N.A.	Yes	

Table III.1 – High-level comparison of ICE modes

	Capability	Full mode	Lite mode	Comment
4	**SIP: negotiation based SDP O/A**			
4.1	eliminating redundant candidates	Yes	No	
4.2	forming check lists (when Offer received)	Yes	No	
5	**Connectivity checks**			
5.1	Initiator role of connectivity checks i.e., STUN role: "STUN client"	Yes	No	
5.2	Responder role of connectivity checks i.e., STUN role: "STUN server"	Yes	Yes	
6	**Completing ICE procedures:**			
6.1	Agent role "ICE controlling"	Yes	No (Note 3)	
6.2	Agent role "ICE controlled"	Yes	Yes	
6.3	Role conflict detection & repair	Yes	No	
6.4	Nominating pairs (regular, agressive)	Yes	No	
6.5	Updating state of candidate pairs	Yes	No	
6.6	Updating state of check lists	Yes	No	
6.7	Freeing candidates	Yes	Yes	
7	**Media handling:**			
7.1	Unconditional start of media sent	Yes	No	[IETF RFC 5245], section 11.1
8	**Interoperability / Extensibility:**			
8.1	Interoperate with full mode	Yes	Yes	
8.2	Interoperate with lite mode	Yes	No	Lite-to-lite = conditional (Note 4)
8.3	Support of ICE extensions	Yes	No	
9	**Performance:**			
9.1	Call setup time (minimal)	Yes (Note 5)	No	Aggressive vs regular nomination
10	**Security:**			
10.1	Security benefits (unrelated to NAT-T)	Yes	No	see Appendix A in [IETF RFC 5245]

Table III.1 – High-level comparison of ICE modes

	Capability	Full mode	Lite mode	Comment
11	**IP network architecture evolution:**			
11.1	Deployment stability	Yes	No (Note 6)	

NOTE 1 – A lite implementation does not gather candidates; it includes only host candidates for any media stream.

NOTE 2 – See Lite (single IPv4 host address, multiple IPv6 addresses; IPv6 connectivity preferred in general (future safe)); References: [IETF RFC 5245], Appendix A "Lite and Full Implementations" and [b-IETF RFC 6724].

NOTE 3 – [IETF RFC 5245], section 5.2: "*For a lite implementation, being the controlling agent means selecting a candidate pair based on the ones in the offer and answer (for IPv4, there is only ever one pair), and then generating an updated offer reflecting that selection, when needed (it is never needed for an IPv4-only host).*"

NOTE 4 – Lite-to-lite represents a theoretical NAT-less end-to-end path, i.e., there would be no NAT traversal required, hence also no ICE required.

NOTE 5 – [IETF RFC 5245]: "*A full implementation will reduce call setup times, since ICE's aggressive mode can be used*."

NOTE 6 – [IETF RFC 5245]: "*it is often the case that a device that finds itself with a public address today will be placed in a network tomorrow where it will be behind a NAT. It is difficult to definitively know, over the lifetime of a device or product, that it will always be used on the public Internet. Full implementation provides assurance that communications will always work*."

The identified capabilities should cover the major aspects. A precise differentiation between both ICE modes would require a formal specification of the complete ICE protocol logic ("which does not exist, neither in IETF nor in other SDOs"). However, in scope of this Recommendation, it should be sufficient to identify ICE mode differences at the level of ITU-T H.248 protocol and procedural specification.

NOTE – Following related items are for further studies:

The remote "ICE agent" could basically provide the same or a different ICE capability set, which raises the question about the principle impact on the MGC or/and MG behaviour?

Appendix IV

ICE extensions: "Trickle ICE"

(This appendix does not form an integral part of this Recommendation.)

IV.1 Introduction

Table IV.1 provides a brief summary of trickle ICE. The information is still preliminary due to the status of the underlying ICE specifications in IETF.

Table IV.1 – Summary "trickle ICE from perspective of ITU-T H.248 gateways"

References:	See: – [b-IETF trickle-ice]: basic ICE protocol procedures and SDP extensions; – [b-IETF trickle-sip]: trickle ICE support via SIP.
Motivation:	– The basic issue with *vanilla ICE*: process of candidate gathering, prioritization and final selection is that it may significantly delay communication establishment ("*relatively lengthy session establishment times and degraded user experience*"). Such delays negatively impact primarily realtime conversation services. – *Trickle ICE*: provides an alternative mode by *incremental* exchange of candidates early during the call establishment process. – ICE agent: additional function "candidate harvester".
Principal impact on main body of this Recommendation:	– Yes: *trickle ICE* is *not backward compatible* to *vanilla ICE*, hence both *ICE agents* (and one is residing in the ITU-T H.248 MG) needs firstly to check *trickle ICE* support or not. – There are again the two variants of *half trickle* and *full trickle* ICE modes. Such asymmetrical options leading to a number of real life combinations between the ITU-T H.248 MG and remote ICE clients (modes of ICE client is either "vanilla", "half trickle" or "full trickle"). – The "trickle ICE" capability declaration and negotiation, as well as fallback process is delegated to the IP signalling plane, i.e., a) the SIP-based MGC call control interface (see clause 6.4.2.1), and b) the ITU-T H.248 gateway control interface (for further studies). Concrete changes are not yet implemented in the ITU-T H.248 capabilities as defined by this Recommendation.

The purpose of this appendix is to illustrate signalling flows with trickle ICE in order to highlight principle impacts on ITU-T H.248 gateway control. The considered use cases should cover the following aspects:

– comparison of signalling scenarios with trickle ICE only and combined trickle/vanilla ICE modes in order to separate trickle ICE specific information at call control signalling level from the remaining signalling part, which is finally visible at the ITU-T H.248 interface at all;

– check impact of trickle ICE on all existing ITU-T H.248 packages as defined by this Recommendation;

– verify possible impact of trickle ICE on the package-less procedures according to clause 10.

The discussion of concrete signalling examples is for further studies.

IV.2 Impact of trickle ICE on ITU-T H.248.50-defined packages

For further studies.

Table IV.2 – Summary "impact of trickle ICE on ITU-T H.248.50-defined packages"

Package	Comments
STUN base package (*stun*) (clause 7.1)	No impact. The address mapping principles as introduced for vanilla ICE should be identical to trickle ICE.
STUN information package (*stuni*) (clause 7.5)	No impact because the NAT type related information is orthogonal to ICE procedures.
MG STUN client package (*mgstunc*) (clause 7.2)	For further studies.
MGC STUN client package (*mgcstunc*) (clause 7.4)	No impact due to MGC-embedded STUN client function, which is not exposed at the ITU-T H.248 interface.
STUN consent freshness package (*stun*) (clause 8.3)	No impact due to the assumption that STUN consent freshness procedures will be requested at a "later point in time" of the call phase, i.e., some time after the "end of address gathering" was already announced
MG TURN client package (*mgstunc*) (clause 7.3)	No impact.
MG act-as STUN server package (*mgastuns*) (clause 8.1)	No impact because the ITU-T H.248-related STUN server assignment is orthogonal to ICE procedures.
Originate STUN continuity check package (*ostuncc*) (clause 8.2)	No impact because STUN continuity checks are a follow-up activity, after the ICE related address gathering phase. There might be additional STUN continuity checks requested (by the MGC from the MG) after an ITU-T H.248 stream is already used for application data transfer.
MGC-originated STUN request package (*mgcostunr*) (clause 9.1)	No impact because keep alive mechanism is orthogonal to ICE procedures. Dynamic view: any keep alive activities occur *after* the ICE based address gathering phase.
Keep alive request package (*mgastuns*) (clause 9.2)	No impact because keep alive mechanism is orthogonal to ICE procedures. Dynamic view: any keep alive activities occur *after* the ICE based address gathering phase.

IV.3 Example signalling flows for trickle ICE

For further studies.

IV.4 Summary

Trickle ICE allows to improve the critical establishment phase which is of key interest of realtime communication services. It is expected that trickle ICE will be subsequently supported and deployed by ICE endpoint implementations. Support of trickle ICE should be revisited by a future release of this Recommendation.

Appendix V

ICE multihomed and IPv4/IPv6 dual stack fairness

(This appendix does not form an integral part of this Recommendation.)

V.1 Introduction

ICE clients could reside in IP host entities with multiple IP interfaces, called multihomed IP hosts. This is the default for ITU-T H.248 IP media gateways when used for network interconnections or/and serving a large number of parallel IP bearer connections. [b-IETF ice-dualstack] describes impact and guidelines for such type of ICE clients.

V.2 Impact on ITU-T H.248 gateways with ICE support?

Support of [b-IETF ice-dualstack] results in *algorithmic changes only* within an ICE entity, but there are *no* syntactical ICE protocol extensions. The technology could be basically supported by the "ITU-T H.248.50 toolkit". There would be changes at all places where the ICE "address prioritization" step is touched, see clauses 6.5.3 and 8.

The MGC or MG could be in charge of the ICE "address prioritization" step, dependent on the underlying network solution related to ICE. When that task is in the MG, then the MG needs to know which algorithm concerning the calculation of priority values is used. This Recommendation does not make any assumptions here. Such information could be e.g., part of an ITU-T H.248 profile specification, also due to the fact that information about the address prioritization algorithm is out of scope of call control signalling (rather a network operator policy).

Appendix VI

ICE restarts

(This appendix does not form an integral part of this Recommendation.)

VI.1 Introduction

At any time during an active communication service, either communication party (ICE endpoints) could restart the process of ICE negotiation. Support of ICE restarts is a basic capability of ICE [IETF RFC 5245]. The possible impact on ITU-T H.248.50-enabled ITU-T H.248 gateways is not yet studied, thus, the present Release of this Recommendation does not yet provide guidelines and specifications for ICE restart procedures.

The purpose of this appendix is to provide a high-level analysis and summary on this particular ICE feature.

VI.2 ICE restart procedures

The starting point is an active call with an active IP bearer connection(s), i.e., there are one or multiple IP transport connections which are all "alive", thus providing a full end-to-end transport connectivity service. ICE restarts might be categorized in three phases:

Phase 1 – possible events which may trigger an ICE restart:

– IP user plane: events related to the conditions of existing connectivity;

– IP signalling plane: events related to call control signalling information (such as an incoming SDP offer with ICE information in case of SIP);

– application and/or call control logic: events related to supplementary services, application control logic (e.g., browser-embedded scripts for updating the application runtime environment, application protocols which explicitly updated transport connectivity, or supplementary service which impact bearer plane routes).

Phase 2 – IP signalling plane:

– call control level ICE restart procedures could be initiated (such as sections 9.1.1.1, 9.2.1.1 and 9.3.1.1 in [IETF RFC 5245] in case of SIP/SDP).

Phase 3 – IP user plane:

– the usual connectivity checks of new, alternative IP transport connections;

– if successful, possibly switchover from existing to new IP transport connection;

– if not, possibly release of the call.

Above three phases reflect the general, high-level scenario in case of ICE restarts. The call association continues, the application data flows continues as well, and the IP transport connection level could provide an uninterrupted transport service at best or a temporary interruption in the range of a loss of a burst of some IP packets.

VI.2 Impact on ITU-T H.248 gateways with ICE support?

There is an impact on ITU-T H.248 gateways in case of ICE restarts. All three ICE restart phases affect the MGC or MG entity or both.

Which ones and how the ITU-T H.248 packages of this Recommendations would be impacted is for further studies.

Bibliography

[b-ITU-T H.248.84] Recommendation ITU-T H.248.84 (2012), *Gateway control protocol: NAT-traversal for peer-to-peer services*.

[b-ITU-T H.248.89] Recommendation ITU-T H.248.89 (2014), *Gateway control protocol: TCP support packages*.

[b-ITU-T H.248.93] Recommendation ITU-T H.248.93 (2014), *Gateway control protocol: H.248 packages for control of transport security using DTLS*.

[b-ITU-T Y.2111] Recommendation ITU-T Y.2111 (2008), *Resource and admission control functions in next generation networks*.

[b-IETF RFC 4571] IETF RFC 4571 (2006), *Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport*.
<https://tools.ietf.org/html/rfc4571>

[b-IETF RFC 5245bis] IETF draft-ietf-ice-rfc5245bis (2015), *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal*.
<https://datatracker.ietf.org/doc/draft-ietf-ice-rfc5245bis/>

[b-IETF RFC 5626] IETF RFC 5626 (2009), *Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)*.
<https://tools.ietf.org/html/rfc5626>

NOTE – Two keep-alive techniques: CRLF keep-alive technique for TCP and SCTP, and STUN keep-alive technique for UDP.

[b-IETF RFC 6223] IETF RFC 6223 (2011), *Indication of Support for Keep-Alive*.
<https://tools.ietf.org/html/rfc6223>

[b-IETF RFC 6724] IETF RFC 6724 (2012), *Default Address Selection for Internet Protocol Version 6 (IPv6)*.
<https://tools.ietf.org/html/rfc6724>

[b-IETF RFC 7362] IETF RFC 7362 (2014), *Latching: Hosted NAT Traversal (HNT) for Media in Real-Time Communication*.
<https://tools.ietf.org/html/rfc7362>

[b-IETF ice-dualstack] IETF draft-ietf-ice-dualstack-fairness (2016), *ICE Multihomed and IPv4/IPv6 Dual Stack Fairness*.
<https://datatracker.ietf.org/doc/draft-ietf-ice-dualstack-fairness/>

[b-IETF ice-lite] IETF draft-rescorla-mmusic-ice-lite (2007), *Implementing Interactive Connectivity Establishment (ICE) in Lite Mode*.
<https://datatracker.ietf.org/doc/draft-rescorla-mmusic-ice-lite/>

[b-IETF ice-sip-sdp] IETF draft-ietf-mmusic-ice-sip-sdp (2009), *Using Interactive Connectivity Establishment (ICE) with Session Description Protocol (SDP) offer/answer and Session Initiation Protocol (SIP)*.
<https://datatracker.ietf.org/doc/draft-ietf-mmusic-ice-sip-sdp/>

[b-IETF no-op] IETF draft-ietf-avt-rtp-no-op (2007), *A No-Op Payload Format for RTP*.
<https://datatracker.ietf.org/doc/draft-ietf-avt-rtp-no-op/>

[b-IETF rkeep] IETF draft-holmberg-sipcore-rkeep (2014), *Indication of support for reverse keep-alive*.
<https://datatracker.ietf.org/doc/draft-holmberg-sipcore-rkeep/>

[b-IETF rtcweb-transports]	IETF draft-ietf-rtcweb-transports (2016), *Transports for WebRTC*. <https://datatracker.ietf.org/doc/draft-ietf-rtcweb-transports/>
[b-IETF trickle-ice]	IETF draft-ietf-ice-trickle (2015), *Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol*. <https://datatracker.ietf.org/doc/draft-ietf-mmusic-trickle-ice/>
[b-IETF trickle-sip]	IETF draft-ietf-mmusic-trickle-ice-sip (2015), *A Session Initiation Protocol (SIP) usage for Trickle ICE*. <https://datatracker.ietf.org/doc/draft-ietf-mmusic-trickle-ice-sip/>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2016