

IPv6 Ready Logo Phase 2  
Session Initiation Protocol

Test Profile  
User Agent  
Endpoint  
Back-to-Back User Agent

Version 2.0.1



## Modification Record

Version 0.1	Jan. 16, 2007	- First release
Ver.0.1.01	Jan. 25, 2007	- Added "Only when a UA supports multiple response." in [REQUIREMENT] of UA-11-1-4 and UA-11-1-9.
Ver.0.1.02	Jan. 26, 2007	- Corrected misspellings.
Ver.0.1.03	Feb. 27, 2007	- Modified the parts that UNH-IOL(Timothy Winters) pointed out. Corrected misspellings.
Ver.1.0.0	Apr. 27, 2007	- Modified the parts that UNH-IOL(Timothy Winters) pointed out. Corrected misspellings.
Ver.1.0.1	Jul. 31, 2007	Version 1.0.1 release. Modified the parts that UNH-IOL (Timothy Winters) and NICI(Fang-Yu Ling) pointed out.
Ver.1.0.2	May. 30, 2008	Remove the scenario UA-4-2-4. Version 1.0.2 release. Modified the points that NICI pointed out. Remove wrong Observable Results. Remove the scenario UA-2-2-5.
Ver.1.1.0	Dec. 12, 2008	Major revision up. (No modification)
Ver.2.0.0	Nov. 27, 2009	- Added Index of BASIC/ADVANCED tests. - Changed [Judgement] into [Observable Results]. - Modified some incorrect parts.
Ver.2.0.1	Jan. 13, 2010	- Modified some misspellings. - Modified some incorrect parts.



# Acknowledgements

IPv6 Forum would like to acknowledge the efforts of the following organizations and commentators in the development of this test specification.

- IPv6 Promotion Council  
Certification Working Group  
SIP IPv6 Sub Working Group

Commentators:

Fang-Yu Ling  
NICI IPv6 Standard and Interoperability Testing Lab.(Telecommunication Laboratories,  
Chunghwa Telecom Co.,Ltd.)



# Table of Contents

[I] IPv6 Ready Logo Phase 2 Test Specification SIP IPv6 User Agent

<b>Modification Record</b> .....	2
<b>Acknowledgements</b> .....	3
<b>Table of Contents</b> .....	4
<b>1 Overview</b> .....	1
<b>2 Requirements of conformance test</b> .....	3
2.1 Requirements based on Policy of SIP IPv6 Ready Logo.....	3
2.2 Other Requirements .....	3
A2.3 Index of BASIC/ADVANCED tests .....	3
2.3.1 Index of BASIC/ADVANCED tests for UA .....	3
2.3.2 Index of BASIC/ADVANCED tests for End Point .....	5
2.3.3 Index of BASIC/ADVANCED tests for B2BUA .....	8
<b>3 Common OBSERVABLE RESULTS</b> .....	10
3.1 generic_message .....	10
3.2 generic_request.....	11
3.3 generic_response.....	12
3.4 generic_SDP .....	13
3.5 generic_200-for-INVITE .....	14
3.6 generic_Initial-INVITE.....	15
3.7 generic_re-INVITE.....	15
3.8 generic_2xx-ACK.....	16
3.9 generic_ACK .....	17
3.10 generic_BYE .....	18
3.11 generic_CANCEL .....	19
3.12 generic_non2xx-ACK .....	22
3.13 generic_REGISTER .....	23
3.14 generic_digest-noqop .....	23
3.15 generic_digest-auth .....	24
<b>4. Test Profile: User Agent operation</b> .....	26
4.1 Registration.....	26
4.1.1 UA-1-1-1 - Successful New Registration .....	26
4.1.2 UA-1-1-2 - Update of Contact List (Refresh).....	31
4.1.3 UA-1-1-4 - Cancellation of Registration.....	37
4.1.4 UA-1-1-5 - Unsuccessful Registration .....	42
4.1.5 UA-1-2-1 - Record-Route in REGISTER response .....	47
4.2 Session Establishment .....	53
4.2.1 UA-2-1-1 - Session Establishment Through Two Proxies (Caller hung up by Callee).....	53
4.2.2 UA-2-1-2 - Session Establishment Through Two Proxies (Callee hung up by Caller) .....	63



4.2.3 UA-2-1-3 - Session Establishment Through Two Proxies (Caller hanging up).....	71
4.2.4 UA-2-1-4 - Session Establishment Through Two Proxies (Callee hanging up).....	77
4.2.5 UA-2-1-5 - Session Establishment Through One Proxy (Caller hung up by Callee)82	
4.2.6 UA-2-1-6 - Session Establishment Through One Proxy (Callee hung up by Caller)92	
4.2.7 UA-2-1-7 - Unsuccessful No Answer [CANCEL] (Caller).....	99
4.2.8 UA-2-1-8 - Unsuccessful No Answer [CANCEL] (Callee).....	107
4.2.9 UA-2-2-1 - Unsuccessful Busy (Caller).....	113
4.2.10 UA-2-2-2 - Unsuccessful Busy (Callee) .....	119
4.2.11 UA-2-2-3 - Unsuccessful No Response from User Agent (Caller).....	123
4.2.12 UA-2-2-4 - Unsuccessful Temporarily Unavailable (Caller) .....	129
4.3 Transaction .....	135
4.3.1 UA-4-1-1 - INVITE Client Transaction (Retransmission) .....	135
4.3.2 UA-4-1-2 - INVITE Client Transaction (Receipt of 180 Ringing and Stop of retransmission).....	140
4.3.3 UA-4-1-3 - INVITE Client Transaction (Stop of ACK upon Timer D fired) .....	143
4.3.4 UA-4-1-4 - Non-INVITE Client Transaction (Stop of retransmission of CANCEL upon Timer F fired).....	147
4.3.5 UA-4-1-5 - Non-INVITE Client Transaction (Stop of retransmission of BYE upon Timer F fired).....	153
4.3.6 UA-4-1-6 - Non-INVITE Client Transaction (Stop of retransmission of REGISTER upon Timer F fired).....	159
4.3.7 UA-4-1-7 - Non-INVITE Client Transaction (Receipt of 100 response to CANCEL and reset of Timer E with T2 ) .....	165
4.3.8 UA-4-1-8 - Non-INVITE Client Transaction (Receipt of 100 response to BYE and reset of Timer E with T2).....	170
4.3.9 UA-4-1-9 - Non-INVITE Client Transaction (Receipt of 100 response to REGISTER and reset of Timer E with T2) .....	174
4.3.10 UA-4-1-10 - INVITE Server Transaction (Stop retransmission after Timer H fired).....	177
4.3.11 UA-4-1-11 - INVITE Server Transaction (Response after Timer H fired) .....	182
4.3.12 UA-4-1-12 - INVITE Server Transaction (Stop of retransmission of 4xx-6xx response upon receipt of ACK).....	187
4.3.13 UA-4-1-13 - Non-INVITE Server Transaction (Stop of retransmission of CANCEL after Timer J fired) .....	190
4.3.14 UA-4-1-14 - Non-INVITE Server Transaction (Stop of retransmission of BYE)...	194
4.3.15 UA-4-2-1 - 487 to CANCEL for INVITE request after 64*T1 fired .....	199
4.3.16 UA-4-2-5 - Time in Expires in INVITE is up (UAS).....	203
4.3.17 UA-4-2-6 - BYE for no response to retransmitted 200 (UAS) .....	206
4.3.18 UA-4-2-7 - No Record-Route in negative replies (caller).....	210
4.3.19 UA-4-2-8 - No Record-Route in negative replies (callee).....	213
4.4 Mid-dialog control .....	218
4.4.1 UA-5-1-1 - Session with re-INVITE (Receiving re-INVITE for Hold) (Caller) .....	218
4.4.2 UA-5-1-2 - Session with re-INVITE (Sending re-INVITE for Hold) (Callee) .....	231
4.4.3 UA-5-2-1 - Receipt of re-INVITE before sending the final response to the first	



INVITE .....	244
4.4.4 UA-5-2-2 - Receipt of re-INVITE before receiving the final response to another re-INVITE .....	250
4.4.5 UA-5-2-3 - Unacceptable re-INVITE (Caller).....	255
4.4.6 UA-5-2-4 - Unacceptable re-INVITE (Callee) .....	259
4.4.7 UA-5-2-5 - No ACK is received for re-INVITE (Caller) .....	264
4.4.8 UA-5-2-6 - No ACK is received for re-INVITE (Callee) .....	268
4.4.9 UA-5-2-7 - re-INVITE without offer (Caller) .....	272
4.4.10 UA-5-2-8 - re-INVITE without offer (Callee).....	277
4.4.11 UA-5-2-9 - Receipt of 491 response to re-INVITE (Caller) .....	281
4.4.12 UA-5-2-10 - Receipt of 491 response for re-INVITE (Callee) .....	288
4.5 Authentication.....	294
4.5.1 UA-6-1-5 - BYE request with user authentication.....	294
4.5.2 UA-6-1-6 - re-INVITE with user authentication .....	298
4.5.3 UA-6-1-7 - REGISTER with Digest Authentication without qop .....	304
4.5.4 UA-6-1-8 - INVITE with Digest Authentication without qop .....	309
4.5.5 UA-6-1-9 - BYE with Digest Authentication without qop .....	315
4.6 Header field parameter.....	322
4.6.1 UA-7-1-1 - Timestamp header field in 100 response .....	322
4.6.2 UA-7-1-2 - URI including a comma, question mark or semicolon .....	326
4.6.3 UA-7-2-1 - Receipt of BYE with an unacceptable header field .....	331
4.6.4 UA-7-2-2 - Receipt of CANCEL with an unacceptable header field .....	335
4.6.5 UA-7-2-3 - Receipt of 200 with an unacceptable header field .....	340
4.7 Routing .....	344
4.7.1 UA-8-1-1 - Proxy performing strict routing .....	344
4.7.2 UA-8-1-2 - Detection of Merged Requests.....	348
4.7.3 UA-8-1-3 - "sent-by" of Via in response not inserted into request.....	353
4.7.4 UA-8-1-4 - "sent-by" in Via with IP address that differs from the packet source address and port .....	358
4.7.5 UA-8-1-5 - Via with "maddr" parameter and port in the "sent-by" .....	364
4.7.6 UA-8-1-6 - Via with "maddr" parameter and without port in "sent-by" .....	370
4.7.7 UA-8-1-7 - "sent-by" in Via with a domain name and a port.....	376
4.7.8 UA-8-1-8 - "sent-by" in Via with a domain name and without a port.....	382
4.7.9 UA-8-1-9 - Correct Strict Routing .....	388
4.8 Request message .....	393
4.8.1 UA-9-2-1 - Unrecognized type of body .....	393
4.8.2 UA-9-2-2 - Unrecognized encoding of body.....	397
4.8.3 UA-9-2-3 - Body in unrecognized language .....	401
4.8.4 UA-9-2-4 - Request with a tag in To header field .....	406
4.8.5 UA-9-2-5 - BYE with value of the lower CSeq .....	409
4.8.6 UA-9-2-6 - Rejection of an offer .....	413
4.8.7 UA-9-2-7 - INVITE with RFC2543 syntax .....	416
4.9 Response message.....	422
4.9.1 UA-10-1-1 - Session Progress response .....	422



4.9.2 UA-10-2-1 - Non-Forwarding of request upon receipt of 503 .....	428
4.9.3 UA-10-2-2 - Receipt of SDP answer in a provisional response.....	431
4.9.4 UA-10-2-3 - Unrecognized response code (2xx).....	435
4.9.5 UA-10-2-4 - Unrecognized response code (4xx).....	441
4.9.6 UA-10-2-5 - Unrecognized response code (5xx).....	444
4.9.7 UA-10-2-6 - Unrecognized response code (6xx).....	448
@te4.9.8 UA-10-2-7 - Provisional response other than a 100 response.....	452
4.9.9 UA-10-2-8 - Omission or abbreviation of body upon receipt of 413(Request Entity Too Large) response .....	457
4.9.10 UA-10-2-9 - Processing request without credentials after receipt of 403 (Forbidden) response.....	461
4.9.11 UA-10-2-10 - Unsupported Require header field.....	464
4.10 Dialog.....	468
4.10.1 UA-11-1-1 - CANCEL for unmatched requests.....	468
4.10.2 UA-11-1-2 - Request without a tag in a From header field.....	473
4.10.3 UA-11-1-3 - Response without a tag in a To header field.....	480
4.10.4 UA-11-1-4 - Multiple 2xx responses .....	486
4.10.5 UA-11-1-5 - Provisional response for the need to ask for an "extension" .....	494
4.10.6 UA-11-1-6 - BYE not matching an existing dialog .....	498
4.10.7 UA-11-1-7 - Write-once Record-routing .....	505
4.10.8 UA-11-1-8 - Returning of correct Record-Route parameter .....	511
4.10.9 UA-11-1-9 - Proper processing upon receipt of multiple 18x from multiple downstream branches .....	517
4.10.10 UA-11-1-10 - Failed re-INVITE not changing the dialog .....	527
4.10.11 UA-11-1-11 - Construction of a correct Route set.....	536
4.11 OPTIONS request.....	544
4.11.1 UA-12-1-1 - Receipt of OPTIONS when the UAS is ready to accept a call .....	544
4.11.2 UA-12-1-2 - Receipt of OPTIONS within a dialog .....	548
4.11.3 UA-12-2-1 - Receipt of OPTIONS when a UAS is busy over a dialog .....	554
4.12 DNS .....	558
4.12.1 UA-13-2-1 - Successful Session with Proxy Failure (Caller) .....	558
4.12.2 UA-13-2-2 - Forwarding of INVITE to an alternative server upon receipt of 503 .....	569
4.13 Transport .....	573
4.13.1 UA-14-2-1 - Receipt of INVITE with additional bytes in the transport packet.....	573
4.13.2 UA-14-2-2 - Transport packet of a response ending before the end of the message body.....	579
4.13.3 UA-14-2-3 - Transport packet of request ending before the end of the message body.....	584
4.14 ICMP.....	589
4.14.1 UA-15-2-1 - Receipt of "ICMP destination unreachable" for a previously sent request.....	589
4.14.2 UA-15-2-2 - Receipt of "ICMP time exceeded" for a previously sent request ....	592
4.14.3 UA-15-2-3 - Receipt of "ICMP time exceeded" for a previously sent response .	595
AUTHORS' LIST .....	600

# 1 Overview

This document describes details of the SIP Conformance Test. The format of the description block is as follows:

## Description block

<b>[NAME]</b>	NAME is a name of the test.
<b>[TARGET]</b>	TARGET is a target node of the test.
<b>[PURPOSE]</b>	PURPOSE is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
<b>[REQUIREMENT]</b>	REQUIREMENT section specifies the functions and conditions that will be needed to perform the test.
<b>[PARAMETER]</b>	PARAMETER describes SIP URIs on the topology that relates to the test
<b>[ADDRESS]</b>	ADDRESS describes IP addresses on the topology that relates to the test.
<b>[TOPOLOGY]</b>	TOPOLOGY describes the network used in the test.
<b>[CONFIGURATION for NUT]</b>	CONFIGURATION for NUT describes how to initialize and configure the NUT before starting each test. If a value is not provided, then the protocol's default value is used.
<b>[INITIALIZATION]</b>	INITIALIZATION describes step-by-step instructions for carrying out the setting before the test.
<b>[PROCEDURE]</b>	PROCEDURE describes step-by-step instructions for carrying out the test.
<b>[OBSERVABLE RESULTS]</b>	<b>OBSERVABLE RESULTS</b> describes expected result. If we can observe as same result as the description of <b>OBSERVABLE RESULTS</b> , the NUT passes the test.
<b>[REFERENCE]</b>	REFERENCE section contains some parts of specification related to the tests. It also shows the document names and section numbers.

NOTE: There are common observable results in the category of OBSERVABLE RESULTS. Refer to Section 3.

## Acronyms

NUT	- Node Under Test (applicant implementation)
IF	- Interface
UNI	- User-Network Interface
NNI	- Network-Network Interface





## Reference standards

- (1) RFC3261: SIP: Session Initiation Protocol (<http://www.ietf.org/rfc/rfc3261.txt>)
- (2) RFC3264: An Offer/Answer Model with Session Description Protocol (<http://www.ietf.org/rfc/rfc3264.txt>)
- (3) RFC4566: SDP: Session Description Protocol (<http://www.ietf.org/rfc/rfc4566.txt>)
- (4) RFC2617: HTTP Authentication: Basic and Digest Access Authentication (<http://www.ietf.org/rfc/rfc2617.txt>)
- (5) RFC3665: SIP Basic Call Flow Examples (<http://www.ietf.org/rfc/rfc3665.txt>)
- (6) IPv6 Ready Logo Phase 2 Policy
- (7) SIP IPv6 Test Scope

## Index

ex . [RFC3261 X.X.X]

Please refer to the table of contents in RFC3261

[RFC3261-X-X]

Please refer to the table number in Test-item-Priority

## 2 Requirements of conformance test

### 2.1 Requirements based on Policy of SIP IPv6 Ready Logo

[PRq]

1. Supported transport protocol is only UDP.
2. The path MTU is 1500 bytes.
3. Supported URI scheme is only SIP-URI.
4. Supported media type of the message-body is application/sdp.
5. Only unicast session is supported.

### 2.2 Other Requirements

[ORq]

1. INVITE requests includes the bodies and any other requests doesn't include the bodies.
2. In case of a tester for conformance test sending message with Record-Route header field.
3. Use the value of Max-Forwards that are configured.
4. Send Digest authentication challenge for establishing a session.
5. In case of B2BUA conformance test, it needs a User Agent for making requests.

## A2.3 Index of BASIC/ADVANCED tests

### 2.3.1 Index of BASIC/ADVANCED tests for UA

Test rank	Function	Test category	Profile No
BASIC	<ul style="list-style-type: none"> <li>- Establishment, disconnection, and cancellation of Session</li> <li>- SDP Offer/Answer (INVITE-200)</li> <li>- Digest authentication (initial INVITE)</li> <li>- Processing of re-INVITE</li> </ul>	Session Establishment	UA-2-1-1
			UA-2-1-2
			UA-2-1-3
			UA-2-1-4
			UA-2-1-5
			UA-2-1-6
			UA-2-1-7
			UA-2-1-8
			UA-2-2-1
			UA-2-2-2
			UA-2-2-3
			UA-2-2-4
		Transaction	UA-4-1-1
			UA-4-1-2
			UA-4-1-3
			UA-4-1-4

			UA-4-1-5 UA-4-1-7 UA-4-1-8 UA-4-1-10 UA-4-1-11 UA-4-1-12 UA-4-1-13 UA-4-1-14 UA-4-2-1 UA-4-2-6 UA-4-2-7 UA-4-2-8
		Mid-dialog control	UA-5-1-1 UA-5-2-1 UA-5-2-3 UA-5-2-4 UA-5-2-5 UA-5-2-6 UA-5-2-7 UA-5-2-8
		Authentication	UA-6-1-8 UA-6-1-9
		Header parameter	UA-7-1-2 UA-7-2-1 UA-7-2-2 UA-7-2-3
		Routing	UA-8-1-2 UA-8-1-3 UA-8-1-4 UA-8-1-5 UA-8-1-6 UA-8-1-7 UA-8-1-8
		Request message	UA-9-2-1 UA-9-2-2 UA-9-2-3 UA-9-2-4 UA-9-2-5 UA-9-2-6
		Response message	UA-10-1-1 UA-10-2-1 UA-10-2-2 UA-10-2-3 UA-10-2-4 UA-10-2-5 UA-10-2-6 UA-10-2-7 UA-10-2-8 UA-10-2-9 UA-10-2-10
		Dialog	UA-11-1-1 UA-11-1-2

			UA-11-1-3 UA-11-1-5 UA-11-1-6 UA-11-1-7 UA-11-1-8 UA-11-1-10 UA-11-1-11
		Transport	UA-14-2-1 UA-14-2-2 UA-14-2-3
		ICMP	UA-15-2-1 UA-15-2-2 UA-15-2-3
ADVANCED	Registration Digest authentication (REGISTER)	Registration	UA-1-1-1 UA-1-1-2 UA-1-1-4 UA-1-1-5 UA-1-2-1 UA-4-1-6 UA-4-1-9 UA-6-1-7
	Digest authentication for BYE	Authentication	UA-6-1-5 UA-6-1-6
	Digest authentication for re-INVITE		
	Hold using re-INVITE	Mid-dialog control	UA-5-1-2 UA-5-2-2 UA-5-2-9 UA-5-2-10
	Configuration of an alternate server	DNS	UA-13-2-1 UA-13-2-2
	Processing of INVITE with Expires header field	Transaction	UA-4-2-5
	Processing of message with RFC2543 syntax	Request message	UA-9-2-7
	Processing of multiple response	Dialog	UA-11-1-4 UA-11-1-9
	Processing of OPTIONS request	OPTIONS method	UA-12-1-1 UA-12-1-2 UA-12-2-1
	Processing of Strict routing	Routing	UA-8-1-1 UA-8-1-9
	Timestamp header field	Header parameter	UA-7-1-1

### 2.3.2 Index of BASIC/ADVANCED tests for End Point

Test rank	Function	Test category	Profile No
BASIC	- Registration - Digest authentication (REGISTER) - Establishment, disconnection, and cancellation of Session	Registration	UA-1-1-1 UA-1-1-2 UA-1-1-4 UA-1-1-5

	<ul style="list-style-type: none"> <li>- SDP Offer/Answer (INVITE-200)</li> <li>- Digest authentication (initial INVITE)</li> <li>- Processing of re-INVITE</li> </ul>		UA-1-2-1 UA-4-1-6 UA-4-1-9 UA-6-1-7
		Session Establishment	UA-2-1-1 UA-2-1-2 UA-2-1-3 UA-2-1-4 UA-2-1-5 UA-2-1-6 UA-2-1-7 UA-2-1-8 UA-2-2-1 UA-2-2-2 UA-2-2-3 UA-2-2-4
		Transaction	UA-4-1-1 UA-4-1-2 UA-4-1-3 UA-4-1-4 UA-4-1-5 UA-4-1-7 UA-4-1-8 UA-4-1-10 UA-4-1-11 UA-4-1-12 UA-4-1-13 UA-4-1-14 UA-4-2-1 UA-4-2-6 UA-4-2-7 UA-4-2-8
		Mid-dialog control	UA-5-1-1 UA-5-2-1 UA-5-2-3 UA-5-2-4 UA-5-2-5 UA-5-2-6 UA-5-2-7 UA-5-2-8
		Authentication	UA-6-1-8 UA-6-1-9
		Header parameter	UA-7-1-2 UA-7-2-1 UA-7-2-2 UA-7-2-3
		Routing	UA-8-1-2 UA-8-1-3 UA-8-1-4 UA-8-1-5 UA-8-1-6 UA-8-1-7

			UA-8-1-8
		Request message	UA-9-2-1 UA-9-2-2 UA-9-2-3 UA-9-2-4 UA-9-2-5 UA-9-2-6
		Response message	UA-10-1-1 UA-10-2-1 UA-10-2-2 UA-10-2-3 UA-10-2-4 UA-10-2-5 UA-10-2-6 UA-10-2-7 UA-10-2-8 UA-10-2-9 UA-10-2-10
		Dialog	UA-11-1-1 UA-11-1-2 UA-11-1-3 UA-11-1-5 UA-11-1-6 UA-11-1-7 UA-11-1-8 UA-11-1-10 UA-11-1-11
		Transport	UA-14-2-1 UA-14-2-2 UA-14-2-3
		ICMP	UA-15-2-1 UA-15-2-2 UA-15-2-3
ADVANCED	Digest authentication for BYE	Authentication	UA-6-1-5
	Digest authentication for re-INVITE		UA-6-1-6
	Hold using re-INVITE	Mid-dialog control	UA-5-1-2 UA-5-2-2 UA-5-2-9 UA-5-2-10
	Configuration of an alternate server	DNS	UA-13-2-1 UA-13-2-2
	Processing of INVITE with Expires header field	Transaction	UA-4-2-5
	Processing of message with RFC2543 syntax	Request message	UA-9-2-7
	Processing of multiple response	Dialog	UA-11-1-4 UA-11-1-9
	Processing of OPTIONS request	OPTIONS method	UA-12-1-1 UA-12-1-2 UA-12-2-1
	Processing of Strict routing	Routing	UA-8-1-1 UA-8-1-9

	Timestamp header field	Header parameter	UA-7-1-1
--	------------------------	------------------	----------

### 2.3.3 Index of BASIC/ADVANCED tests for B2BUA

Test rank	Function	Test category	Profile No
BASIC	<ul style="list-style-type: none"> <li>- Establishment, disconnection, and cancellation of Session</li> <li>- SDP Offer/Answer (INVITE-200)</li> <li>- Digest authentication (initial INVITE)</li> <li>- Processing of re-INVITE</li> </ul>	Session Establishment	UA-2-1-1
			UA-2-1-2
			UA-2-1-3
			UA-2-1-4
			UA-2-1-5
			UA-2-1-6
			UA-2-1-7
			UA-2-1-8
			UA-2-2-1
			UA-2-2-2
			UA-2-2-3
			UA-2-2-4
		Transaction	UA-4-1-1
			UA-4-1-2
			UA-4-1-3
			UA-4-1-4
			UA-4-1-5
			UA-4-1-7
			UA-4-1-8
			UA-4-1-10
			UA-4-1-11
			UA-4-1-12
			UA-4-1-13
			UA-4-1-14
			UA-4-2-1
			UA-4-2-6
			UA-4-2-7
			UA-4-2-8
		Mid-dialog control	UA-5-1-1
			UA-5-1-2
			UA-5-2-1
			UA-5-2-2
			UA-5-2-3
			UA-5-2-4
			UA-5-2-5
			UA-5-2-6
			UA-5-2-7
			UA-5-2-8
			UA-5-2-9
			UA-5-2-10
		Authentication	UA-6-1-8
			UA-6-1-9
		Header parameter	UA-7-1-2
			UA-7-2-1
			UA-7-2-2
			UA-7-2-3

		Routing	UA-8-1-2 UA-8-1-3 UA-8-1-4 UA-8-1-5 UA-8-1-6 UA-8-1-7 UA-8-1-8
		Request message	UA-9-2-1 UA-9-2-2 UA-9-2-3 UA-9-2-4 UA-9-2-5 UA-9-2-6
		Response message	UA-10-1-1 UA-10-2-1 UA-10-2-2 UA-10-2-3 UA-10-2-4 UA-10-2-5 UA-10-2-6 UA-10-2-7 UA-10-2-8 UA-10-2-9 UA-10-2-10
		Dialog	UA-11-1-1 UA-11-1-2 UA-11-1-3 UA-11-1-5 UA-11-1-6 UA-11-1-7 UA-11-1-8 UA-11-1-10 UA-11-1-11
		Transport	UA-14-2-1 UA-14-2-2 UA-14-2-3
		ICMP	UA-15-2-1 UA-15-2-2 UA-15-2-3
ADVANCED	Registration Digest authentication (REGISTER)	Registration	UA-1-1-1 UA-1-1-2 UA-1-1-4 UA-1-1-5 UA-1-2-1 UA-4-1-6 UA-4-1-9 UA-6-1-7
	Digest authentication for BYE	Authentication	UA-6-1-5
	Digest authentication for re-INVITE		UA-6-1-6
	Configuration of an alternate server	DNS	UA-13-2-1 UA-13-2-2
	Processing of INVITE with Expires	Transaction	UA-4-2-5



	header field		
	Processing of message with RFC2543 syntax	Request message	UA-9-2-7
	Processing of multiple response	Dialog	UA-11-1-4 UA-11-1-9
	Processing of OPTIONS request	OPTIONS method	UA-12-1-1 UA-12-1-2 UA-12-2-1
	Processing of Strict routing	Routing	UA-8-1-1 UA-8-1-9
	Timestamp header field	Header parameter	UA-7-1-1

## 3 Common OBSERVABLE RESULTS

### 3.1 generic\_message

Generic observable results for SIP message

The size of whole message must be less than or equal to 1500 bytes. [PRq-1,2]

- The empty line must be present even if the message-body is not. [RFC3261-7-2]
- Request-Line and Status-Line:
  - Must exist as a start-line. [RFC3261 7]
  - Must be terminated by a carriage-return line-field sequence (CRLF). [RFC3261-7-1]
  - SIP-version: MUST be "SIP/2.0". [RFC3261-7-5,6]
- Header field:
  - Must be terminated by a carriage-return line-field sequence (CRLF). [RFC3261-7-1]
  - Recommended that Via, Route, Record-Route, Proxy-Require, Max-Forwards, and Proxy-Authorization appear towards the top of the message to facilitate rapid parsing. [RFC3261-7-7]
- \* To
  - addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-22][RFC3261 20.10]
- \* From
  - addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-22][RFC3261-20-34]
- \* Content-Length



The Content-Length header SHOULD be present, and the value SHOULD equal the size of the message-body, in decimal number of octets. [RFC3261-20-28]

## 3.2 generic\_request

Generic observable results for SIP request

- Request-Line:

Request-URI:

Must not contain unescaped spaces or control characters. [RFC3261-7-3]

Must not be enclosed in "<>". [RFC3261-7-4]

- Header fields:

\* To

Must exist. [RFC3261-8-1]

addr-spec: Must be the specified SIP-URI as UA1(AoR). (Excepting REGISTER request.) [RFC3261 8.1.1.2][RFC3261 20.39]

\* From

Must exist. [RFC3261-8-1]

addr-spec: Must be the specified SIP-URI as NUT(AoR). [RFC3261 8.1.1.3][RFC3261 20.20]

tag-param: Must exist. [RFC3261-8-9, RFC3261-12-37]

\* Call-ID

Must exist. [RFC3261-8-1]

\* CSeq

Must exist. [RFC3261-8-1]

Must be less than 2\*\*31. [RFC3261-8-15,16, RFC3261-20-32]

method: Must match that of the request. [RFC3261-8-14, RFC3261-9-4, RFC3261-12-44, RFC3261-13-21, RFC3261-17-37]

\* Max-Forwards

Must exist. [RFC3261-8-1, RFC3261-8-17]

Must be the value specified in the tester configuration. [RFC3261-8-18]

\* Via

Must exist. [RFC3261-8-1, RFC3261-8-19]

via-branch: Must exist in each Via header field. [RFC3261-8-21,55]

via-branch:

token: Must be different from all values sent by NUT in this sequence.



(Excepting ACK for non-2xx response and CANCEL.) [RFC3261-8-22]

via-branch:

token: Must begin with the characters "z9hG4bK". [RFC3261-8-23, RFC3261-20-46]

transport: Must be "UDP". [PRq -1]

sent-by:

host: Recommended being specified hostname of NUT. [RFC3261-18-12]

protocolname and protocol version:

Must be SIP/2.0 [RFC3261-8-20]

### 3.3 generic\_response

Generic observable results for SIP response

- Status-Line:

Status-Code: Must exist. [RFC3261 7.2]

Status-Code: Must be three digit integer. [RFC3261 7.2]

- Header fields:

\* From

Must exist. [RFC3261-8-98]

Must equal that of the request. (Tester check the value of URI and tag, respectively.) [RFC3261-8-98]

\* Call-ID

Must exist. [RFC3261-8-99]

Must equal that of the request. [RFC3261-8-10,99]

\* CSeq

Must exist. [RFC3261-8-100]

Must equal that of the request. [RFC3261-8-100]

\* Via

Must exist. [RFC3261-8-101, 102]

via-branch: Must exist in each Via header field. [RFC3261-8-21]

via-param: Values must equal those in the request. [RFC3261-8-101]

via-param: Values must maintain the same order as those in the request. (Tester check the value of sent-by, via-branch, and via-received of expect 1st line, respectively.) [RFC3261-8-47, 102]

\* To

Must exist. [RFC3261-8-103, 104, 105]



Must equal that of the request if the request contained a tag-param. (Tester check the value of URI and tag, respectively.) [RFC3261-8-103]

addr-spec: Must equal that in the request if the request did not contain a tag-param. [RFC3261-8-104]

tag-param: Must be added if the request did not contain a tag-param. (Excepting 100 response.) [RFC3261-8-105]

### 3.4 generic\_SDP

Generic observable results for SDP

The lines of v, o, s, c\*, t, media-part must appear in this order. (Items marked with a "\*" may be omitted.) [RFC3264-5-1][RFC4566-5-4]

The lines of m, c\*, a\* must appear in this order in the media part. (Items marked with a "\*" may be omitted.) [RFC3264-5-1, RFC4566-5-4]

If a c line in the session-part is omitted, all media-parts must include a c line. [RFC3264-5-1, RFC4566-5-29]

Only one SDP body Must exist. [RFC3264-5-4]

\* "v=" line v=0

Must be "0". [RFC3264-5-1][RFC4566 5.1]

\* "o=" line o=<username> <sess-id> <sess-version> <nettype> <addrtype>  
<unicast-address>

<sess-id>: Must be a 64 bit signed integer. [RFC3264-5-2,3]

<sess-version>: Must be a 64 bit signed integer. [RFC3264-5-2,3]

<nettype>: Must be "IN". [RFC3264-5-1][RFC4566 5.2]

<addrtype>: Must be "IP6". [RFC3264-5-1][RFC4566 5.2]

<unicast-address>: Must be specified hostname or global unicast address of NUT.  
[RFC3264-5-1][RFC4566 5.2]

\* "s=" line s=<session name>

<session name>: Recommended to be a single space character (0x20) or a dash (-).  
[RFC3264-5-5][RFC4566 5.3]

\* "c=" line c=<nettype> <addrtype> <connection-address>

<nettype>: Must be "IN". [RFC3264-5-1][RFC4566 5.7]

<addrtype>: Must be "IP6". [RFC3264-5-1][RFC4566 5.7]

<connection-address>: Must be the global unicast address of NUT.  
[RFC3264-5-1][RFC4566 5.7]

\* "t=" line t=<start-time> <stop-time>

Should be "0 0" if the SDP is an offer. [RFC3264-5-6]



Must equal that of the offer if the SDP is an answer. [RFC3264-6-5]

\* "m=" line m=<media> <port> <proto> <fmt list>

Must contain the same number of "m=" lines as the offer if the SDP is an answer.  
[RFC3264-6-3][RFC4566 5.14]

\* "a=" line a=<attribute>, a=<attribute>:<value>

If the attribute is "ptime", the value must be greater than zero.  
[RFC3264-5-18][RFC4566 5.13]

### 3.5 generic\_200-for-INVITE

Generic observable results for 200 response to the Initial INVITE

- Header fields:

\* Contact

Must exist. [RFC3261-12-4]

addr-spec: Must be the specified SIP-URI as NUT(Contact Address).  
[RFC3261-12-5, PRq-3]

addr-spec:

host: Recommended being specified hostname of NUT. [RFC3261-19-3]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is  
contained. [RFC3261-20-22]

\* Content-Type

Must exist. [RFC3261-7-14, RFC3261-20-31, RFC3261-13-7]

media-type: Must be "application/sdp". [PRq-4]

\* Allow

Should exist. [RFC3261-13-36]

\* Supported

Should exist. [RFC3261-13-36]

(option-tag: Must be corresponding to standards-track RFCs, however the tester  
does not judge this.) [RFC3261-8-28]

- Bodies:

Must exist. [RFC3261-13-7,13,37]



### 3.6 generic\_Initial-INVITE

Generic observable results for Initial INVITE

- Request-Line:

Method: Must be "INVITE". [RFC3261 7.1]

Request-URI: Must be the same value as the addr-spec in the To header field.  
[RFC3261-8-2]

- Header fields:

\* To

tag-param: Must not exist. [RFC3261-8-7]

\* Contact

Must exist. [RFC3261-8-24, RFC3261-12-20]

addr-spec: Must be the specified SIP-URI as NUT(Contact Address).  
[RFC3261-12-20, PRq-3]

addr-spec:

host: Recommended being specified hostname of NUT. [RFC3261-19-3]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is  
contained. [RFC3261-20-22]

\* Content-Type

Must exist. [RFC3261-7-14, RFC3261-20-31]

media-type: Must be "application/sdp". [PRq -4]

\* Allow

Should exist. [RFC3261-13-2]

\* Supported

Should exist. [RFC3261-13-4]

(option-tag: Must be corresponding to standards-track RFCs, however the tester  
does not judge this.) [RFC3261-8-28]

- Bodies:

Must exist. [ORq-1][RFC3261-13-6,13]

### 3.7 generic\_re-INVITE

Generic observable results for re-INVITE

- Request-Line:



Method: Must be "INVITE". . [RFC3261 7.1]

- Header fields:

\* To

tag-param: Must exist. [RFC3261-12-35]

\* Call-ID

call-id: Must equal that of the Initial INVITE. [RFC3261-8-10, RFC3261-12-40]

\* CSeq

sequence number: Must be the value incremented by one from that of the previous request sent by NUT, if any. [RFC3261-12-41,42,43]

\* Contact

Should exist. [RFC3261-12-52]

addr-spec: Should be the specified SIP-URI as NUT(Contact Address).  
[RFC3261-12-53]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-22]

\* Content-Type

Must exist. [RFC3261-20-31]

media-type: Must be "application/sdp". [PRq-4]

- Bodies:

Must exist. [ORq-1]

### 3.8 generic\_2xx-ACK

Generic observable results for ACK, which is correspond to 200 OK response to INVITE request.

- Request-Line:

Method: Must be "ACK". . [RFC3261 7.1]

Request-URI: Must equal that Contact header field value in the 200 OK response.  
[RFC3261-12-47]

- Header fields:

\* To

tag-param: Must exist. [RFC3261-12-35]



tag-param: Must equal that contained in the To header field of 200 response.  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of INVITE request.  
[RFC3261-12-37]

\* Call-ID

call-id: Must be set to the Call-ID of the dialog. [RFC3261-8-10, RFC3261-12-40]

\* CSeq

sequence number: MUST be the same as the INVITE being acknowledged.  
[RFC3261-13-20]

\* Proxy-Authorization

MUST contain the same credentials as the INVITE. [RFC3261-13-22]

\* Require

May exist.

option-tag: Must contain only those Require values that were present in the initial request. [RFC3261-8-82]

\* Proxy-Require

May exist.

option-tag: Must contain only those Proxy-Require values that were present in the initial request. [RFC3261-8-82]

- Bodies:

Must not exist. [ORq-1]

### 3.9 generic\_ACK

Generic observable results for ACK

- Header fields:

\* Accept

Must not exist. [RFC3261-20-8]

\* Accept-Encoding

Must not exist. [RFC3261-20-8]

\* Accept-Language





- Must not exist. [RFC3261-20-8]
- \* Alert-Info
  - Must not exist. [RFC3261-20-8]
- \* Allow
  - Must not exist. [RFC3261-20-8]
- \* Expires
  - Must not exist. [RFC3261-20-8]
- \* In-Reply-To
  - Must not exist. [RFC3261-20-8]
- \* Organization
  - Must not exist. [RFC3261-20-8]
- \* Priority
  - Must not exist. [RFC3261-20-8]
- \* Reply-To
  - Must not exist. [RFC3261-20-8]
- \* Server
  - Must not exist. [RFC3261-20-8]
- \* Subject
  - Must not exist. [RFC3261-20-8]
- \* Supported
  - Must not exist. [RFC3261-20-8]
- \* Warning
  - Must not exist. [RFC3261-20-8]

### 3.10 generic\_BYE

- Request-Line:
  - Method: Must be "BYE". . [RFC3261 7.1]
- Header fields:
  - \* To



tag-param: Must exist. [RFC3261-12-35]

\* Call-ID

call-id: Must equal that of the Initial INVITE. [RFC3261-8-10, RFC3261-12-40]

\* CSeq

sequence number: Must be the value incremented by one from that of the previous request sent by NUT, if any. [RFC3261-12-41,42,43]

\* Alert-Info

Must not exist. [RFC3261-20-8]

\* Contact

Must not exist. [RFC3261-20-8]

\* Expires

Must not exist. [RFC3261-20-8]

\* In-Reply-To

Must not exist. [RFC3261-20-8]

\* Organization

Must not exist. [RFC3261-20-8]

\* Priority

Must not exist. [RFC3261-20-8]

\* Reply-To

Must not exist. [RFC3261-20-8]

\* Subject

Must not exist. [RFC3261-20-8]

\* Warning

Must not exist. [RFC3261-20-8]

- Bodies:

Must not exist. [ORq-1]

### 3.11 generic\_CANCEL

The destination address, port, and transport

Must be equal those used to send the original request. [RFC3261-9-11]



- Request-Line:

Method: Must be "CANCEL". . [RFC3261 7.1]

Request-URI: Must be equal that in the request being cancelled. [RFC3261-9-2]

- Header fields:

\* Call-ID

Must equal that in the request being cancelled. [RFC3261-9-2]

\* To

Must equal that in the request being cancelled, including a tag. (Tester check URI and tag, respectively.)[RFC3261-9-2]

\* CSeq

sequence number: Must equal that in the request being cancelled. [RFC3261-9-2]

\* From

Must equal that in the request being cancelled, including a tag. (Tester check the value of URI and tag, respectively.)[RFC3261-9-2]

\* Via

MUST have only a single Via header field value. [RFC3261-9-3]

Must be the value matching the top Via value in the request being cancelled. (Tester check the value of sent-by, via-branch, and via-received, respectively.)[RFC3261-9-3]

\* Route

Must exist if the request being cancelled contains a Route header field. [RFC3261-9-5]

\* Accept

Must not exist. [RFC3261-20-8]

\* Accept-Encoding

Must not exist. [RFC3261-20-8]

\* Accept-Language

Must not exist. [RFC3261-20-8]

\* Alert-Info

Must not exist. [RFC3261-20-8]



- \* Allow
  - Must not exist. [RFC3261-20-8]
- \* Contact
  - Must not exist. [RFC3261-20-8]
- \* Content-Disposition
  - Must not exist. [RFC3261-20-8]
- \* Content-Encoding
  - Must not exist. [RFC3261-20-8]
- \* Content-Language
  - Must not exist. [RFC3261-20-8]
- \* Expires
  - Must not exist. [RFC3261-20-8]
- \* In-Reply-To
  - Must not exist. [RFC3261-20-8]
- \* MIME-Version
  - Must not exist. [RFC3261-20-8]
- \* Organization
  - Must not exist. [RFC3261-20-8]
- \* Priority
  - Must not exist. [RFC3261-20-8]
- \* Proxy-Authorization
  - Must not exist. [RFC3261-20-8]
- \* Proxy-Require
  - Must not exist. [RFC3261-9-6,RFC3261-20-8,RFC3261-8-80]
- \* Reply-To
  - Must not exist. [RFC3261-20-8]
- \* Require
  - Must not exist. [RFC3261-9-6,RFC3261-20-8,RFC3261-8-80]
- \* Subject



Must not exist. [RFC3261-20-8]

- Bodies:

Must not exist. [ORq-1]

### 3.12 generic\_non2xx-ACK

- Request-Line:

Method: Must be "ACK". . [RFC3261 7.1]

Request-URI: Must be the same value of that in the original request.  
[RFC3261-17-32]

- Header fields:

\* From

Must be the same value of that in the original request. [RFC3261-17-32]

\* Call-ID

Must be the same value of that in the original request. [RFC3261-8-10,  
RFC3261-17-32]

\* To

Must equal the To header field in the response being acknowledged.  
[RFC3261-17-33]

\* Via

MUST be equal to the top Via header field of the original request. [RFC3261-17-34,  
35]

\* CSeq

sequence number: Must be the same value as was present in the original request.  
[RFC3261-17-36]

\* Require

Must not exist. [RFC3261-8-80,RFC3261-20-8]

\* Proxy-Require

Must not exist. [RFC3261-8-80,RFC3261-20-8]

- Bodies:

Must not exist. [ORq-1]



### 3.13 generic\_REGISTER

Generic observable results for REGISTER

- Request-Line:

Method: Must be "REGISTER". [RFC3261 7.1]

Request-URI: Must be specified URI as REGISTRAR. [RFC3261-10-5]

SIP-URI: Must not include userinfo and "@". [RFC3261-10-6]

- Header fields:

\* To

addr-spec: Must be the specified SIP-URI as NUT(AoR). [RFC3261-10-7,PRq-3]

tag: Must not exist. [RFC3261-8-7]

\* Contact

action: Should not exist. [RFC3261-10-11]

\* Record-Route

Must not exist. [RFC3261-20-8]

\* Alert-Info

Must not exist. [RFC3261-20-8]

\* In-Reply-To

Must not exist. [RFC3261-20-8]

\* Priority

Must not exist. [RFC3261-20-8]

\* Reply-To

Must not exist. [RFC3261-20-8]

\* Subject

Must not exist. [RFC3261-20-8]

- Bodies:

Must not exist. [ORq-1]

### 3.14 generic\_digest-noqop

Generic observable results for Authorization and Proxy-Authorization header field



-Header fields:

- \* Authorization
- \* Proxy-Authorization

credentials: Must begin with "Digest". [RFC2617 3.2.1]  
username: Must exist. [RFC2617 3.2.2]  
username-value: Must be set to the specified value. [RFC2617 3.2.2]  
realm: Must exist. [RFC2617 3.2.2]  
realm-value: Must equal that of the WWW-Authenticate header field in 401 response  
or the Proxy-Authenticate header field in 407 response. [RFC2617 3.2.2,  
3.6]  
nonce: Must exist. [RFC2617 3.2.2]  
nonce-value: Must equal that of the WWW-Authenticate header field in 401 response  
or the Proxy-Authenticate header field in 407 response. [RFC2617 3.2.2,  
3.6]  
message-qop: Must not exist. [RFC2617 3.2.2, rfc2069]  
digest-uri: Must exist. [RFC2617 3.2.2]  
digest-uri-value: Must be enclosed in quotation marks. [RFC3261-22-34]  
digest-uri-value: Must equal Request-URI. [RFC3261 22.4, RFC2617 3.2.2]  
dresponse: Must exist. [RFC2617 3.3.2]  
request-digest: Must be calculated in the manner of RFC2617 3.2.2.(MD5/no  
qop)[RFC2617 3.2.2]

### 3.15 generic\_digest-auth

-Header fields:

- \* Authorization
- \* Proxy-Authorization

credentials: Must begin with "Digest". [RFC2617 3.2.1]  
username: Must exist. [RFC2617 3.2.2]  
username-value: Must be set to the specified value. [RFC2617 3.2.2]  
realm: Must exist. [RFC2617 3.2.2]  
realm-value: Must equal that of the WWW-Authenticate header field in 401 response  
or the Proxy-Authenticate header field in 407 response. [RFC2617  
3.2.2, 3.6]  
nonce: Must exist. [RFC2617 3.2.2]  
nonce-value: Must equal that of the WWW-Authenticate header field in 401 response  
or the Proxy-Authenticate header field in 407 response. [RFC2617  
3.2.2.1]



digest-uri: Must exist. [RFC2617 3.2.2]  
digest-uri-value: Must be enclosed in quotation marks. [RFC3261-22-34]  
digest-uri-value: Must equal Request-URI. [RFC3261 22.4, RFC2617 3.2.2]  
message-qop: Must exist. [RFC3261-22-37, RFC2617 3.2.1]  
qop-value: Must be set to "auth". [RFC3261-22-37]  
nonce-count: Must exist. [RFC3261-22-37, RFC2617 3.3.2]  
cnonce: Must exist. [RFC3261-22-37, RFC2617 3.3.2]  
dresponse: Must exist. [RFC2617 3.3.2]  
request-digest: Must be calculated in the manner of RFC2617 3.2.2.(MD5/qop=auth)  
[RFC2617 3.2.2]



## 4. Test Profile: User Agent operation

### 4.1 Registration

#### 4.1.1 UA-1-1-1 - Successful New Registration

**[NAME]**

UA-1-1-1 - Successful New Registration

**[PURPOSE]**

Verify that a NUT properly creates a new registration

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

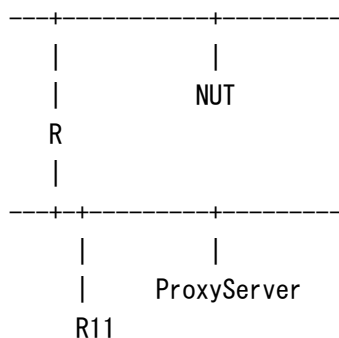
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

**[TOPOLOGY]**

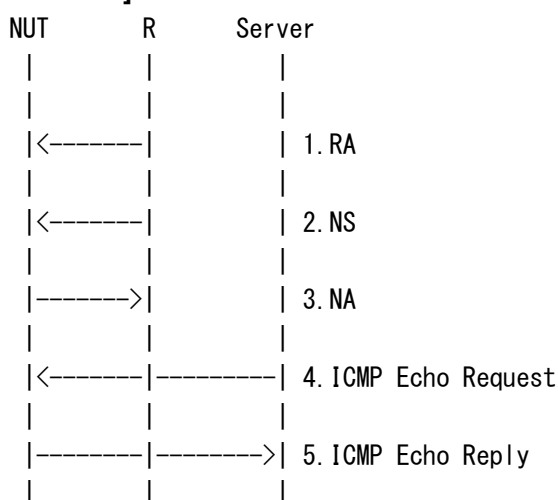




# **[CONFIGURATION for NUT]**

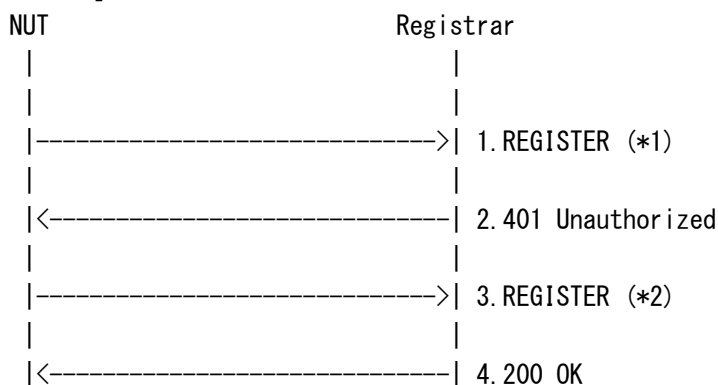
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

# **[INITIALIZATION]**



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# **[PROCEDURE]**



1. Receive REGISTER. (\*1)
2. Send 401 Unauthorized.
3. Receive REGISTER. (\*2)
4. Send 200 OK.

**=== Message example ===**

**1. REGISTER NUT -> Registrar**

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 1 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Content-Length: 0

**2. 401 Unauthorized Registrar -> NUT**

SIP/2.0 401 Unauthorized  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl  
To: NUT <sip:NUT@under.test.com>;tag=1410948204  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 1 REGISTER  
WWW-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

**3. REGISTER NUT -> Registrar**

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlfIH  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600



Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",  
qop=auth, nc=00000002, cnonce="d4e4cec0",  
uri="sip:reg.under.test.com",  
response="b7fd380421adc89263e6774026cfc049"  
Content-Length: 0

#### 4. 200 OK Registrar -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlf1H  
To: NUT <sip:NUT@under.test.com>;tag=37GkEhwl6  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>;expires=3600  
Date: Sat,13 Nov 2004 23:29:00 GMT  
Content-Length: 0

#### [OBSERVABLE RESULTS]

\*1:REGISTER request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_REGISTER

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_REGISTER

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is  
contained. [RFC3261-20-22]

contact-param: Must be the Contact address of NUT. [RFC3261.10.2.1]



contact-param: "\*" MUST NOT be used unless the Expires header field is present with a value of "0". [RFC3261-10-15]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact header field does not exist). [RFC3261.10.2.2]

- Bodies:

See generic\_REGISTER

\*2: REGISTER request from NUT

As a SIP Message,

See generic\_request

As a SIP request,

- Request-Line:

See generic\_request

See generic\_REGISTER

- Header fields:

See generic\_request

- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: Must be the same as that of \*1 message. [RFC3261-8-11] [RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from \*1 message by one with the same Call-ID. [RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-22]

contact-param: Must be the Contact address of NUT. [RFC3261.10.2.1]

contact-param: "\*" MUST NOT be used unless the Expires header field is present with a value of "0". [RFC3261-10-15]

c-p-expires:



delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact header field does not exist). [RFC3261.10.2.2]

\* Authorization

Should exist. [RFC3261-22-16]

See generic\_digest-auth

- Bodies:

See generic\_REGISTER

**[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Section 2.1

## 4.1.2 UA-1-1-2 - Update of Contact List (Refresh)

**[NAME]**

UA-1-1-2 - Update of Contact List (Refresh)

**[PURPOSE]**

Verify that a NUT properly processes update of Contact List by refreshing.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

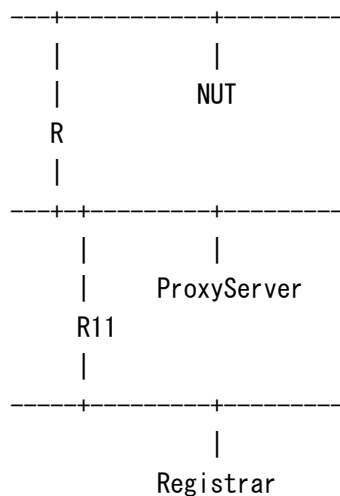
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

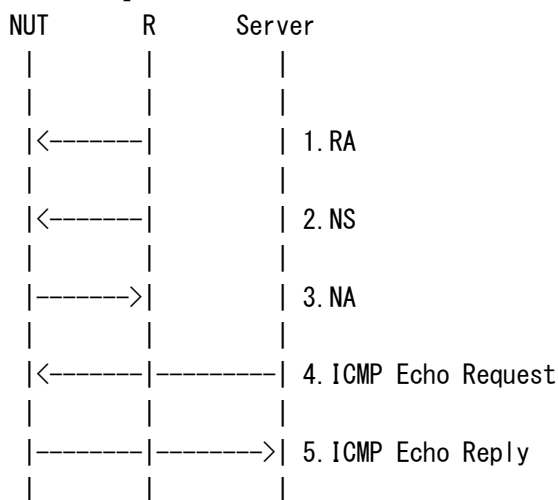
## [TOPOLOGY]



## [CONFIGURATION for NUT]

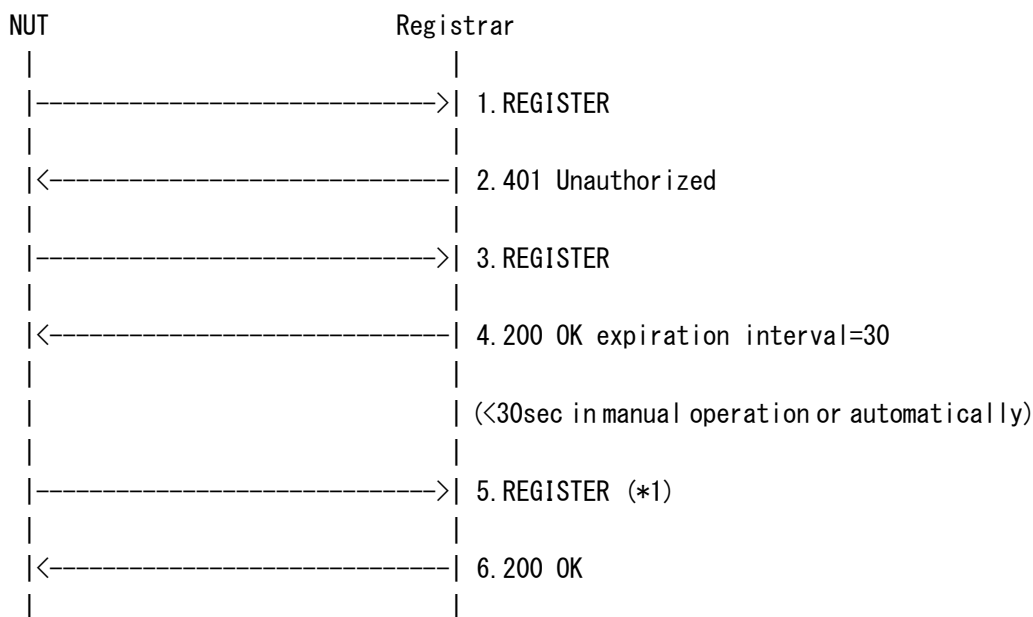
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

## [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]



1. Receive REGISTER.
2. Send 401 Unauthorized.
3. Receive REGISTER.
4. Send 200 OK.
5. Receive REGISTER. (\*1)
6. Send 200 OK.

**=== Message example ===**

**1. REGISTER NUT -> Registrar**

```
REGISTER sip:reg.under.test.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl
To: NUT <sip:NUT@under.test.com>
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com
CSeq: 1 REGISTER
Contact: <sip:NUT@node.under.test.com>
Expires: 3600
Content-Length: 0
```

**2. 401 Unauthorized Registrar -> NUT**

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7
;received=3ffe:501:ffff:5::X
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl
To: NUT <sip:NUT@under.test.com>;tag=1410948204
```





Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 1 REGISTER  
WWW-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. REGISTER NUT -> Registrar

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlfH  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",  
qop=auth, nc=00000002, cnonce="d4e4cec0",  
uri="sip:reg.under.test.com",  
response="b7fd380421adc89263e6774026cfc049"  
Content-Length: 0

### 4. 200 OK Registrar -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlfH  
To: NUT <sip:NUT@under.test.com>;tag=37GkEhw16  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>;expires=30  
Date: Sat,13 Nov 2004 23:29:00 GMT  
Content-Length: 0

### 5. REGISTER NUT -> Registrar

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq  
To: NUT <sip:NUT@under.test.com>



Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 3 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Authorization: Digest username="NUT",  
    realm="under.test.com",  
    nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",  
    qop=auth, nc=00000002, cnonce="d4e4cec0",  
    uri="sip:reg.under.test.com",  
    response="b7fd380421adc89263e6774026cfc049"  
Content-Length: 0

#### 6. 200 OK Registrar -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9  
    ;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq  
To: NUT <sip:NUT@under.test.com>;tag=34095828jh  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 3 REGISTER  
Contact: <sip:NUT@node.under.test.com>;expires=3600  
Date: Sat, 13 Nov 2004 23:29:30 GMT  
Content-Length: 0

#### [OBSERVABLE RESULTS]

- \*1 sends within 30 seconds in manual operation or automatically,
  - This is the Test for checking that UA sends "update REGISTER",  
    so whichever is sufficient even if it sends in manual operation or automatically.

\*1:REGISTER request from NUT.

As a SIP Message,  
    See generic\_message

As a SIP request,

- Request-Line:  
    See generic\_request  
    See generic\_REGISTER  
    Request-URI: Must be the same URI as that of 3. REGISTER. [RFC3261-10-18]
- Header fields:  
    See generic\_request



- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: Must be the same as that of 1st/2nd REGISTER. [RFC3261-8-11]  
[RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from 2nd REGISTER request by one with the  
same Call-ID. [RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is  
contained. [RFC3261-20-22]

contact-param: Must be the Contact address of NUT. [RFC3261.10.2.1]

contact-param: "\*" MUST NOT be used unless the Expires header field is  
present with a value of "0". [RFC3261-10-15]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact  
header field does not exist). [RFC3261.10.2.2]

\* Authorization

Should exist. [RFC3261-22-16]

See generic\_digest-auth

- Bodies:

See generic\_REGISTER

## [REFERENCE]

RFC3665 SIP Basic Call Flow Examples

See Section 2.2

[rfc3261]

### 10.2.1.1 Setting the Expiration Interval of Contact Addresses

When a client sends a REGISTER request, it MAY suggest an expiration  
interval that indicates how long the client would like the



registration to be valid. (As described in Section 10.3, the registrar selects the actual time interval based on its local policy.)

contact-param: "\*" MUST NOT be used unless the Expires header field is present with a value of "0". [RFC3261-10-15]

### 4.1.3 UA-1-1-4 - Cancellation of Registration

#### [NAME]

UA-1-1-4 - Cancellation of Registration

#### [PURPOSE]

Verify that a NUT properly cancels a registration.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

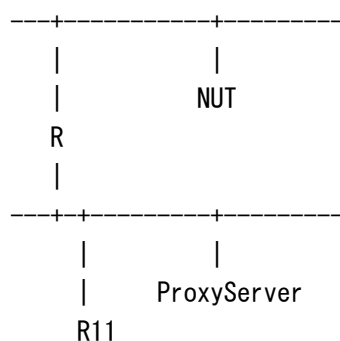
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

#### [TOPOLOGY]

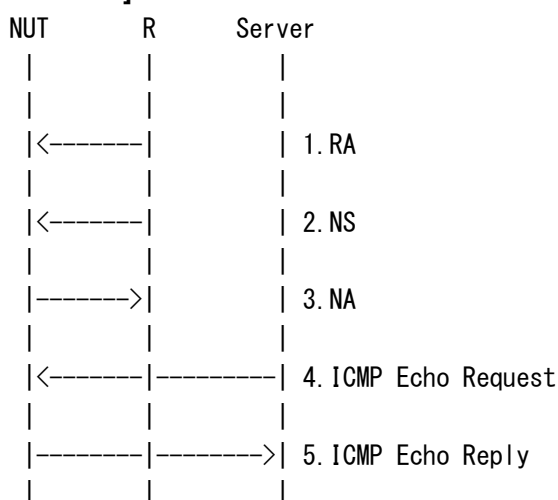




# [CONFIGURATION for NUT]

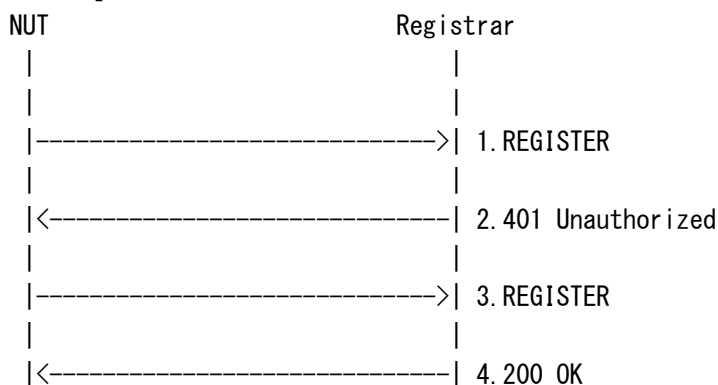
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

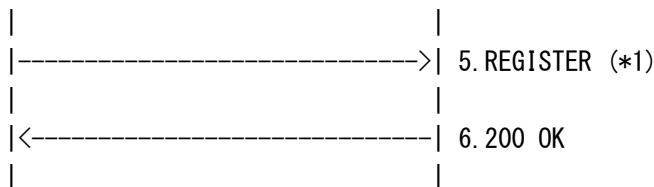
# [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]





1. Receive REGISTER.
2. Send 401 Unauthorized.
3. Receive REGISTER.
4. Send 200 OK.
5. Receive REGISTER (\*1).
6. Send 200 OK.

**=== Message example ===**

**1. REGISTER NUT -> Registrar**

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 1 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Content-Length: 0

**2. 401 Unauthorized Registrar -> NUT**

SIP/2.0 401 Unauthorized  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl  
To: NUT <sip:NUT@under.test.com>;tag=1410948204  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 1 REGISTER  
WWW-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

**3. REGISTER NUT -> Registrar**

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
Max-Forwards: 70



From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlfH  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Authorization: Digest username="NUT",  
    realm="under.test.com",  
    nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",  
    qop=auth, nc=00000002, cnonce="d4e4cec0",  
    uri="sip:reg.under.test.com",  
    response="b7fd380421adc89263e6774026cfc049"  
Content-Length: 0

#### **4. 200 OK Registrar -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
    ;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=ja743ks76zlfH  
To: NUT <sip:NUT@under.test.com>;tag=37GkEhw16  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Contact: <sip:NUT@node.under.test.com>;expires=3600  
Date: Sat,13 Nov 2004 23:28:50 GMT  
Content-Length: 0

#### **5. REGISTER NUT -> Registrar**

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 3 REGISTER  
Expires: 0  
Contact: \*  
Authorization: Digest username="NUT",  
    realm="under.test.com",  
    nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",  
    qop=auth, nc=00000002, cnonce="d4e4cec0",  
    uri="sip:reg.under.test.com",  
    response="b7fd380421adc89263e6774026cfc049"  
Content-Length: 0



## 6. 200 OK Registrar -> NUT

SIP/2.0 200 OK

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9

;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq

To: NUT <sip:NUT@under.test.com>;tag=1418nmdsrf

Call-ID: 1j9FpLxk3uxtm8tn@under.test.com

CSeq: 3 REGISTER

Date: Sat,13 Nov 2004 23:29:00 GMT

Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:REGISTER request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_REGISTER

- Header fields:

See generic\_request

- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: The same value for registrations must be sent to a particular registrar.  
[RFC3261-8-11] [RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from 2nd REGISTER request by one with the  
same Call-ID. [RFC3261-10-9]

\* Contact

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is  
contained. [RFC3261-20-22]

Either of the following conditions must be fulfilled.

- 1) Contact address with URI equal to Contact URI of NUT is present, and  
the Expires header field with a value of "0" is present, and no Contact  
header field with a value of "\*" is present. [RFC3261-10-14]





- 2) Only one Contact header field with a value of "\*" is present, and the Expires header field with a value of "0" is present. [RFC3261-10-15]

\* Authorization  
Should exist. [RFC3261-22-16]  
See generic\_digest-auth

- Bodies:  
See generic\_REGISTER

#### [REFERENCE]

RFC3665 SIP Basic Call Flow Examples  
See Section 2.4

### 4.1.4 UA-1-1-5 - Unsuccessful Registration

#### [NAME]

UA-1-1-5 - Unsuccessful Registration

#### [PURPOSE]

Verify that a NUT properly processes an unsuccessful registration.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

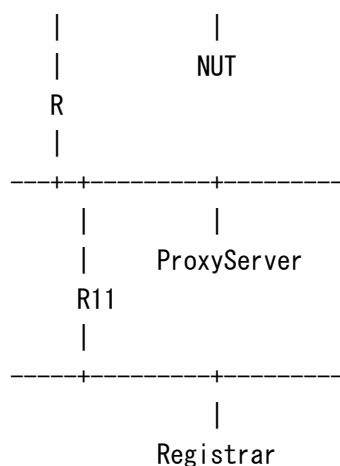
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

#### [TOPOLOGY]

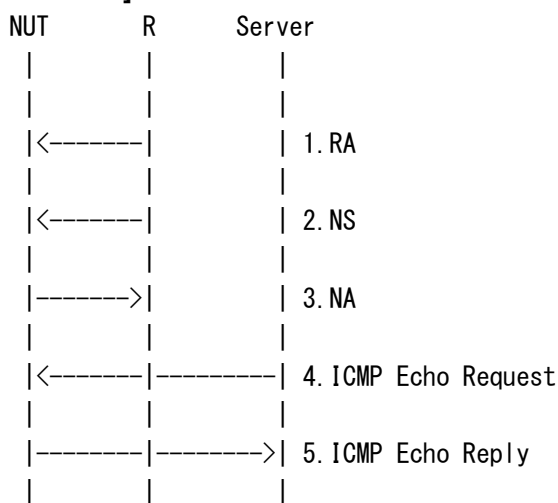
-----+-----+-----



### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

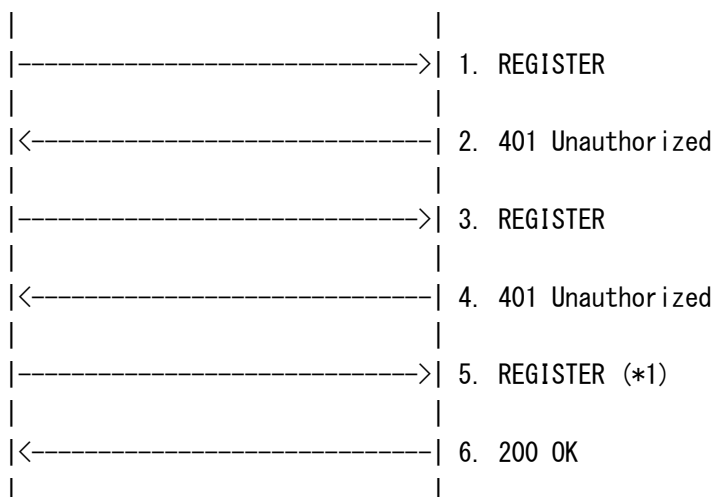
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Receive REGISTER.
2. Send 401 Unauthorized.
3. Receive REGISTER.
4. Send 401 Unauthorized.
5. Receive REGISTER. (\*1)
6. Send 200 OK.

**=== Message example ===**

**1. REGISTER NUT -> Registrar**

```
REGISTER sip:reg.under.test.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl
To: NUT <sip:NUT@under.test.com>
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com
CSeq: 1 REGISTER
Contact: <sip:NUT@node.under.test.com>
Expires: 3600
Content-Length: 0
```

**2. 401 Unauthorized Registrar -> NUT**

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7
;received=3ffe:501:ffff:5::X
From: NUT <sip:NUT@under.test.com>;tag=a73kszlfl
To: NUT <sip:NUT@under.test.com>;tag=1410948204
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="under.test.com", qop="auth",
```



nonce="f1cec4341ae6ca9c8e88df84be55a359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. REGISTER NUT -> Registrar

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=JueHGuidj28dfga  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f1cec4341ae6ca9c8e88df84be55a359", opaque="",  
qop=auth, nc=00000002, cnonce="d4e4cec0",  
uri="sip:reg.under.test.com",  
response="b7fd380421adc89263e6774026cfc049"  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Content-Length: 0  
/\* The registrar changes nonce value \*/

### 4. 401 Unauthorized Registrar -> NUT

SIP/2.0 401 Unauthorized  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds8  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=JueHGuidj28dfga  
To: NUT <sip:NUT@under.test.com>;tag=8jfMowR4  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 2 REGISTER  
WWW-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="84f1c1ae6cbe5ua9c8e88dfa3ecm3459",  
opaque="", stale=TRUE, algorithm=MD5  
Content-Length: 0

### 5. REGISTER NUT -> Registrar

REGISTER sip:reg.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com



CSeq: 3 REGISTER  
Contact: <sip:NUT@node.under.test.com>  
Expires: 3600  
Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="84f1c1ae6cbe5ua9c8e88dfa3ecm3459", opaque="",  
qop=auth, nc=00000002, cnonce="d4e4cec0",  
uri="sip:reg.under.test.com",  
response="1cec4341ae6cbe5a359ea9c8e88df84f"  
Content-Length: 0

#### 6. 200 OK Registrar -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=jS8Mwi2mq  
To: NUT <sip:NUT@under.test.com>;tag=1418nmdsrf  
Call-ID: 1j9FpLxk3uxtm8tn@under.test.com  
CSeq: 3 REGISTER  
Contact: <sip:NUT@node.under.test.com>;expires=3600  
Date: Sat,13 Nov 2004 23:29:00 GMT  
Content-Length: 0

#### [OBSERVABLE RESULTS]

\*1:REGISTER request from NUT.

As a SIP Message,  
See generic\_message  
As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_REGISTER

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_REGISTER

\* Call-ID

callid: The same value for registrations must be sent to a particular registrar.  
[RFC3261-8-11] [RFC3261-10-8, 17]



\* Cseq

1\*DIGIT: Must be incremented from 2nd REGISTER request by one with the same Call-ID. [RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-22]

contact-param: "\*" MUST NOT be used unless the Expires header field is present with a value of "0". [RFC3261-10-15]

Must be the Contact address of NUT. [RFC3261.10.2.1]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact header field does not exist).[RFC3261.10.2.2]

\* Authorization

Should exist. [RFC3261-22-16]

See generic\_digest-auth

- Bodies:

See generic\_REGISTER

**[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Section 2.5

### **4.1.5 UA-1-2-1 - Record-Route in REGISTER response**

**[NAME]**

UA-1-2-1 - Record-Route in REGISTER response

**[PURPOSE]**

Verify that a NUT ignores a Record-Route header field in a REGISTER response.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

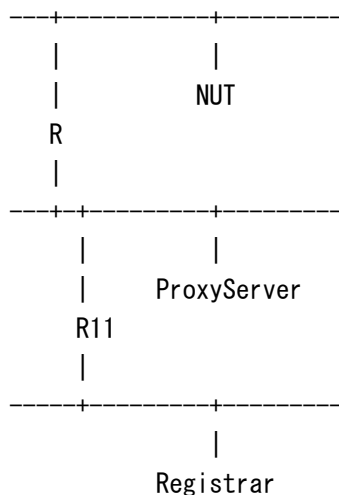
# [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Proxy0	reg-proxy.under.test.com

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

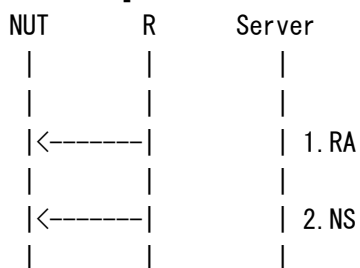
# [TOPOLOGY]

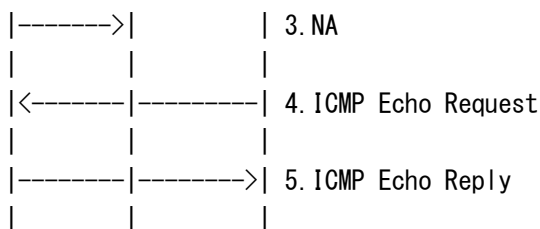


# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

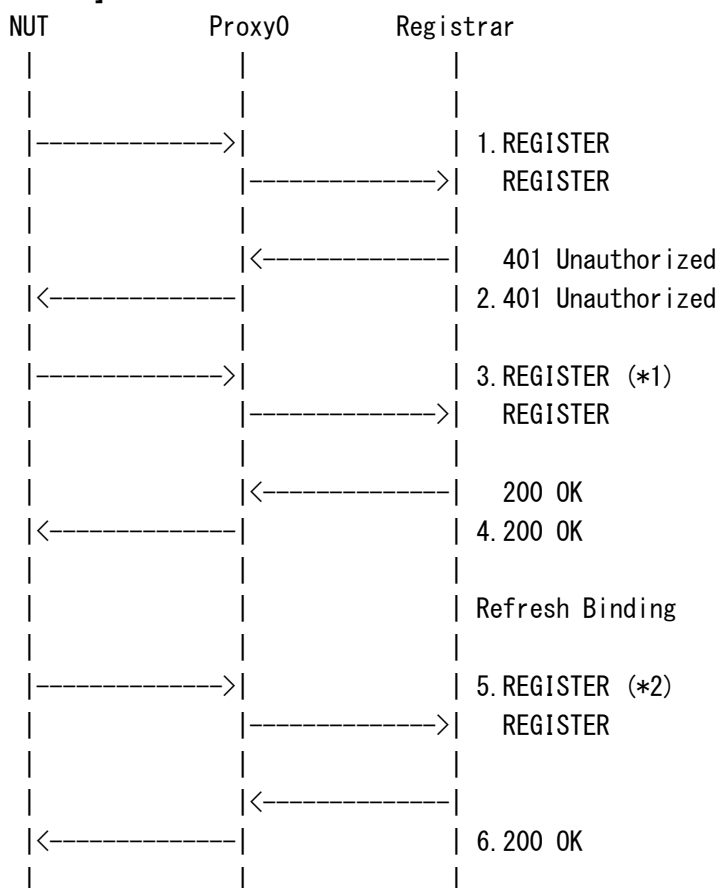
# [INITIALIZATION]





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive REGISTER.
2. Send 401 Unauthorized.
3. Receive REGISTER. (\*1)
4. Send 200 OK.
5. Receive REGISTER. (\*2)





6. Send 200 OK.

=== Message example ===

## 2. 401 Unauthorized Proxy0 -> NUT

SIP/2.0 401 Unauthorized

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X

Record-Route: <sip:reg-proxy.under.test.com:5060;lr>

From: LittleGuy <sip:NUT@under.test.com>;tag=a73kszlfl

To: LittleGuy <sip:NUT@under.test.com>;tag=1410948204

Call-ID: 1j9FpLxk3uxtm8tn@under.test.com

CSeq: 1 REGISTER

WWW-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",  
opaque="", stale=FALSE, algorithm=MD5

Content-Length: 0

## 4. 200 OK Proxy0 -> NUT

SIP/2.0 200 OK

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X

From: LittleGuy <sip:NUT@under.test.com>;tag=ja743ks76zlfIH

To: LittleGuy <sip:NUT@under.test.com>;tag=37GkEhw16

Record-Route: <sip:reg-proxy.under.test.com:5060;lr>

Call-ID: 1j9FpLxk3uxtm8tn@under.test.com

CSeq: 2 REGISTER

Contact: <sip:NUT@node.under.test.com>;expires=30

Date: Sat, 13 Nov 2004 23:29:00 GMT

Content-Length: 0

## [OBSERVABLE RESULTS]

\*1: REGISTER request from NUT

As a SIP Message,

See generic\_request

As a SIP request,

- Request-Line:

See generic\_request

See generic\_REGISTER

Request-URI: Must be equal to that of "1.REGISTER". [RFC3261-10-3,  
RFC3261-10-4]

- Header fields:



See generic\_request

- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: Must be the same as that of \*1 message. [RFC3261-8-11]  
[RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from \*1 message by one with the same  
Call-ID. [RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question  
mark is contained. [RFC3261-20-22]

contact-param: "\*" MUST NOT be used unless the Expires header field is  
present with a value of "0". [RFC3261-10-15]

contact-param: Must be the Contact address of NUT. [RFC3261.10.2.1]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact  
header field does not exist). [RFC3261.10.2.2]

\* Authorization

Should exist. [RFC3261-22-16]

See generic\_digest-auth

\* Route

Must not exist. [RFC3261-10-3, RFC3261-10-4]

- Bodies:

See generic\_REGISTER

\*2: REGISTER request from NUT

As a SIP Message,

See generic\_request

As a SIP request,

- Request-Line:



See generic\_request

See generic\_REGISTER

Request-URI: Must be equal to that of "1.REGISTER". [RFC3261-10-3, RFC3261-10-4]

- Header fields:

See generic\_request

- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: Must be the same as that of \*1 message. [RFC3261-8-11]  
[RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from \*1 message by one with the same  
Call-ID. [RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question  
mark is contained. [RFC3261-20-22]

contact-param: "\*" MUST NOT be used unless the Expires header field is  
present with a value of "0". [RFC3261-10-15]

contact-param: Must be the Contact address of NUT. [RFC3261.10.2.1]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact  
header field does not exist). [RFC3261.10.2.2]

\* Authorization

Should exist. [RFC3261-22-1]

See generic\_digest-auth

\* Route

Must not exist. [RFC3261-10-3, RFC3261-10-4]

- Bodies:

See generic\_REGISTER

## [REFERENCE]

[rfc3261]

[RFC3261-10-3, RFC3261-10-4]

### 10.2 Constructing the REGISTER Request

A REGISTER request does not establish a dialog. A UAC MAY include a Route header field in a REGISTER request based on a pre-existing route set as described in Section 8.1. The Record-Route header field has no meaning in REGISTER requests or responses, and MUST be ignored if present. In particular, the UAC MUST NOT create a new route set based on the presence or absence of a Record-Route header field in any response to a REGISTER request.

## 4.2 Session Establishment

### 4.2.1 UA-2-1-1 - Session Establishment Through Two Proxies (Caller hung up by Callee)

#### [NAME]

UA-2-1-1 - Session Establishment Through Two Proxies (Caller hung up by Callee)

#### [PURPOSE]

Verify that a NUT (caller) properly processes an established session on two proxies when the callee hangs up.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

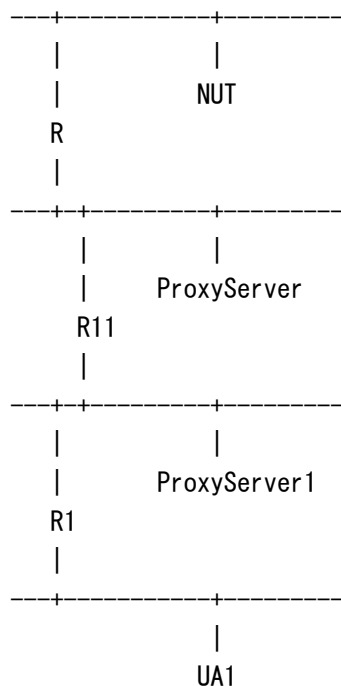
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

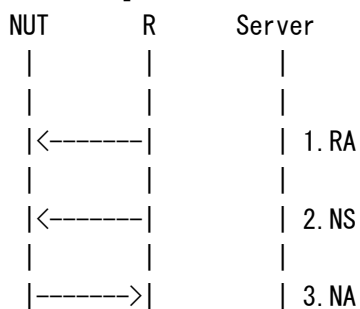
#### [TOPOLOGY]

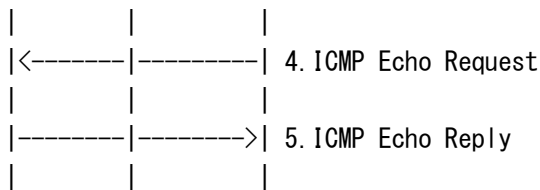


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

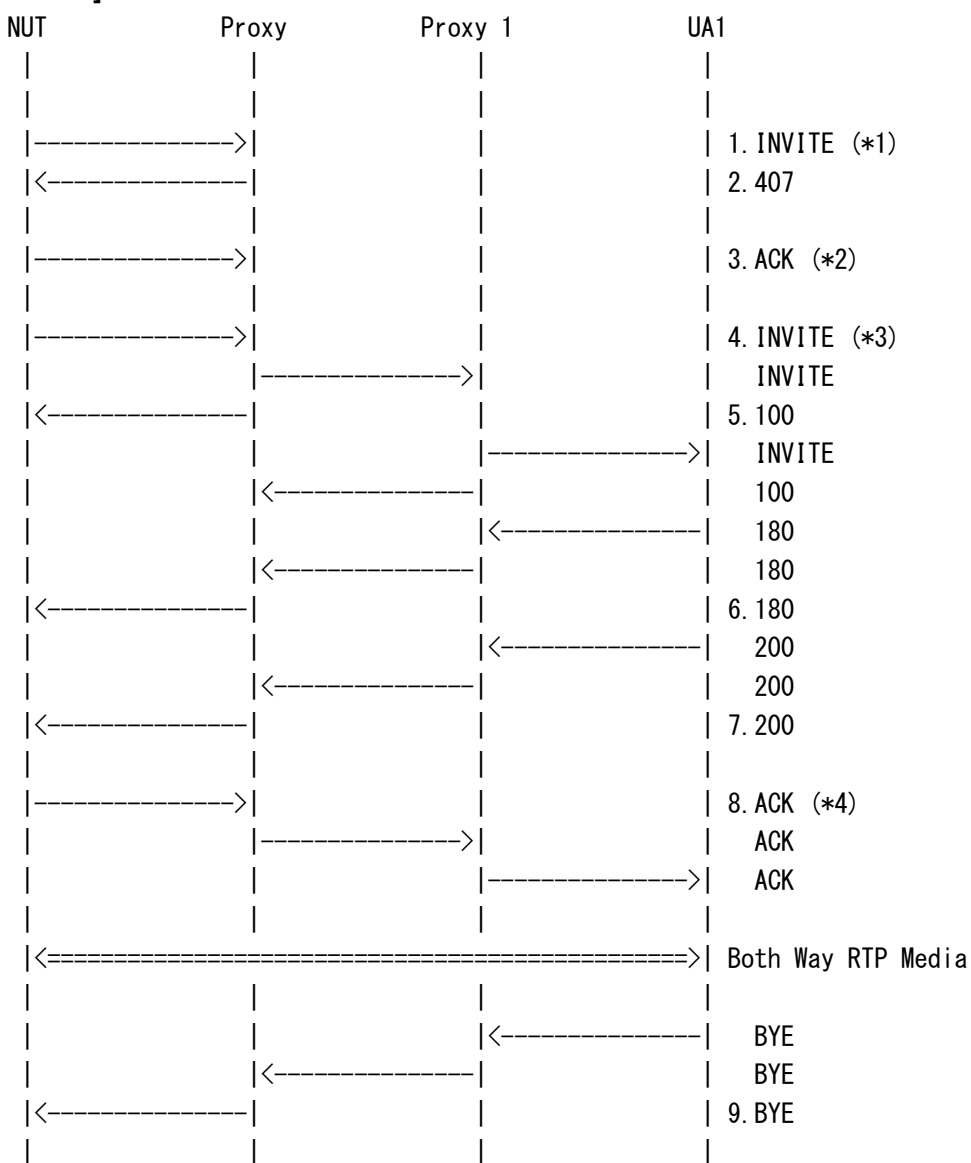
#### [INITIALIZATION]

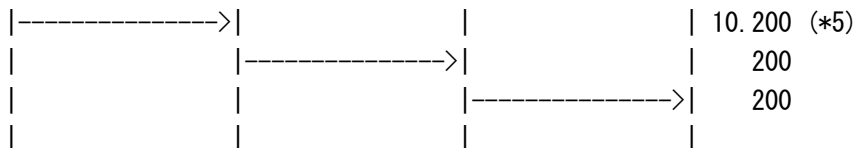




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE. (\*1)
2. Send 407 Proxy Authorization Required.
3. Receive ACK. (\*2)
4. Receive INVITE. (\*3)
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK. (\*4)
9. Send BYE.
10. Receive 200 OK. (\*5)

**=== Message example ===**

**1. INVITE NUT -> Proxy**

INVITE sip:UA1@atlanta.example.com SIP/2.0  
 Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
 Route: <sip:ss.under.test.com;lr>  
 Max-Forwards: 70  
 From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
 To: UA1 <sip:UA1@atlanta.example.com>  
 Call-ID: 3848276298220188511@under.test.com  
 CSeq: 1 INVITE  
 Contact: <sip:NUT@node.under.test.com>  
 Supported:  
 Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
 Content-Type: application/sdp  
 Content-Length: 151

v=0  
 o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
 s=-  
 c=IN IP6 3ffe:501:ffff:5::X  
 t=0 0  
 m=audio 49172 RTP/AVP 0  
 a=rtpmap:0 PCMU/8000  
 /\* Proxy challenges NUT for authentication \*/

**2. 407 Proxy Authorization Required Proxy -> NUT**

SIP/2.0 407 Proxy Authorization Required



Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="f84f1cec41e6cbe5aea9c8e88d359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 ACK  
Content-Length: 0

/\* NUT re-sends INVITE with authentication  
credentials in it. \*/

### 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp





Content-Length: 151

v=0

o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X

s=-

c=IN IP6 3ffe:501:ffff:5::X

t=0 0

m=audio 49172 RTP/AVP 0

a=rtpmap:0 PCMU/8000

/\* Proxy accepts the credentials and forwards the INVITE to Proxy

2. Client for NUT prepares to receive data on port 49172 from the network. \*/

### 5. 100 Trying Proxy -> NUT

SIP/2.0 100 Trying

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

Call-ID: 3848276298220188511@under.test.com

CSeq: 2 INVITE

Content-Length: 0

### 6. 180 Ringing Proxy -> NUT

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X

Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 3848276298220188511@under.test.com

CSeq: 2 INVITE

Contact: <sip:UA1@client.atlanta.example.com>

Content-Length: 0

### 7. 200 OK Proxy -> NUT

SIP/2.0 200 OK

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X

Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl



To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### **8. ACK NUT -> Proxy**

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b76  
Route: <sip:ss.under.test.com;lr>,  
      <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
      realm="under.test.com",  
      nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
      qop=auth, nc=00000004, cnonce="6f54a149",  
      uri="sip:UA1@atlanta.example.com",  
      response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

#### **9. BYE Proxy -> NUT**

BYE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
      ;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
      ;received=3ffe:501:ffff:1::1  
Max-Forwards: 68



From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

#### 10. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

#### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*2:ACK request from NUT.



As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

\*3:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

\* Call-ID

callid: Should be the same as that of "1.INVITE" request, with case-sensitivity on. [RFC3261-8-62]



\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

See generic\_digest-auth

- Bodies:

See generic\_Initial-INVITE

See generic\_SDP

\*4:ACK request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*5:200 response from NUT.



As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response

\* Status-Code: Must be "200". [RFC3261.22.2.1]

- Header fields:  
See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

#### [REFERENCE]

RFC3665 SIP Basic Call Flow Examples

See Section 3.2

### 4.2.2 UA-2-1-2 - Session Establishment Through Two Proxies (Callee hung up by Caller)

#### [NAME]

UA-2-1-2 - Session Establishment Through Two Proxies (Callee hung up by Caller)

#### [PURPOSE]

Verify that a NUT (callee) properly processes an established session on two proxies when the caller hangs up.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

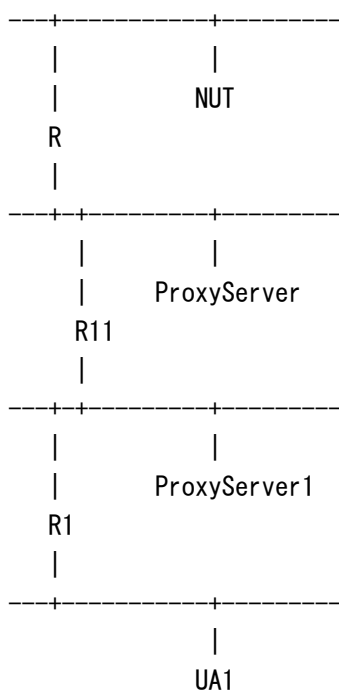
NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

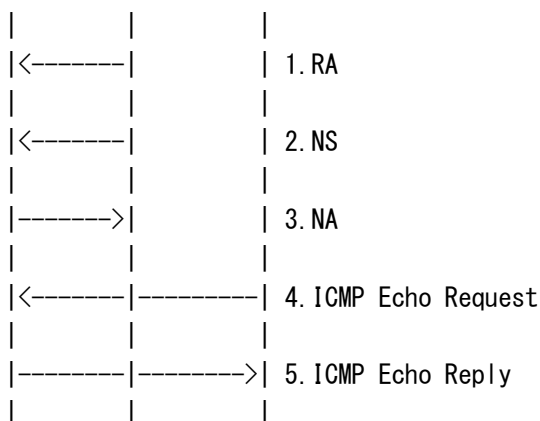


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

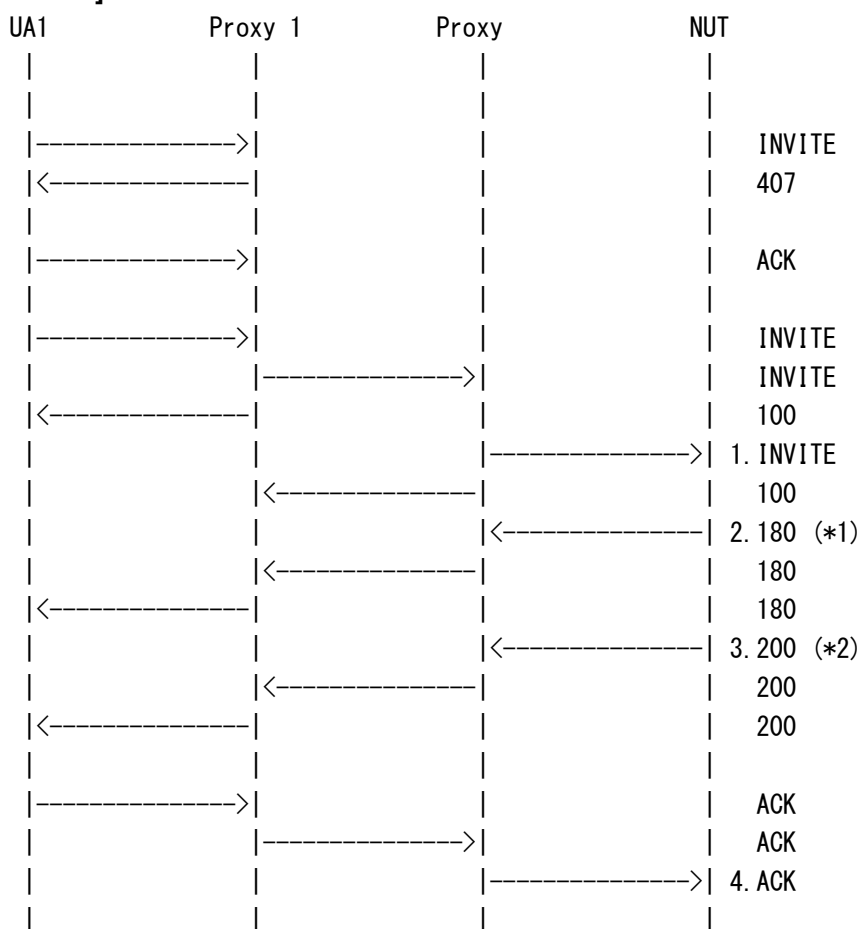
#### [INITIALIZATION]



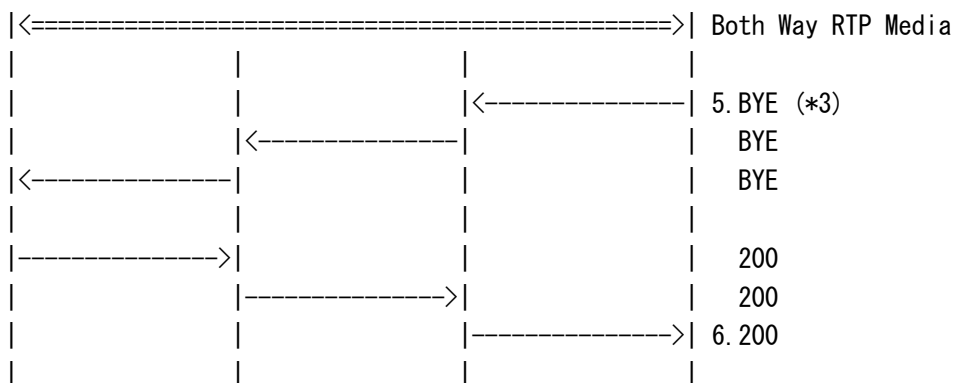


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE. (\*3)
6. Send 200 OK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
<sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Supported:
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Type: application/sdp
Content-Length: 151

```

```

v=0
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1

```



t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

### **2.180 Ringing NUT -> Proxy**

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Length: 0

### **3.200 OK NUT -> Proxy**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=NUT 2890844527 2890844527 IN IP6 3ffe:501:ffff:5::X



s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### **4.ACK Proxy -> NUT**

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d5810.2  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74b76  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 ACK  
Content-Length: 0  
/\* RTP streams are established between UA1 and NUT \*/  
/\* NUT Hangs Up with UA1. \*/  
/\* Again, note that the CSeq is NOT 3. UA1 and NUT maintain  
their own separate CSeq counts \*/

#### **5.BYE NUT -> Proxy**

BYE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
Max-Forwards: 70  
Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 1 BYE  
Content-Length: 0

#### **6.200 OK Proxy -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com



CSeq: 1 BYE

Content-Length: 0

### **[OBSERVABLE RESULTS]**

\*1:180 response from NUT. [Optional]

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:

See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

See generic\_200-for-INVITE

Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]



- Header fields:

See generic\_response

- outside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

\*3:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request. [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE".



[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "3.200" response.  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

#### [REFERENCE]

Sequence in Section 3.7., RFC3665

### 4.2.3 UA-2-1-3 - Session Establishment Through Two Proxies (Caller hanging up)

#### [NAME]

UA-2-1-3 - Session Establishment Through Two Proxies (Caller hanging up)

#### [PURPOSE]

Verify that a NUT (caller) properly processes an established session on two proxies when the caller hangs up.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

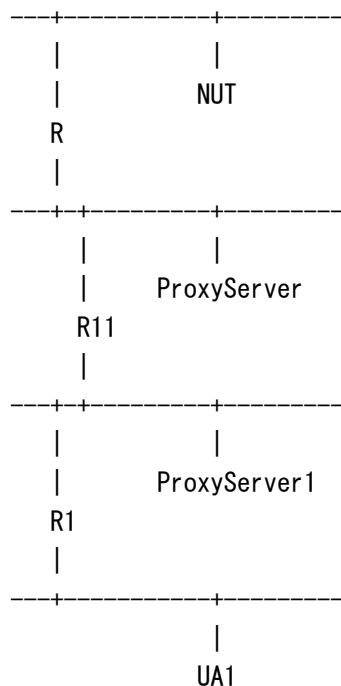
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

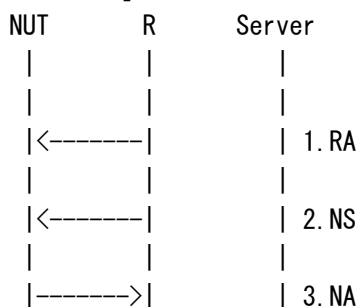
#### [TOPOLOGY]

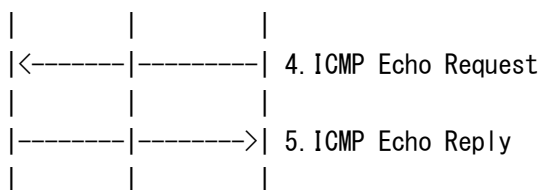


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

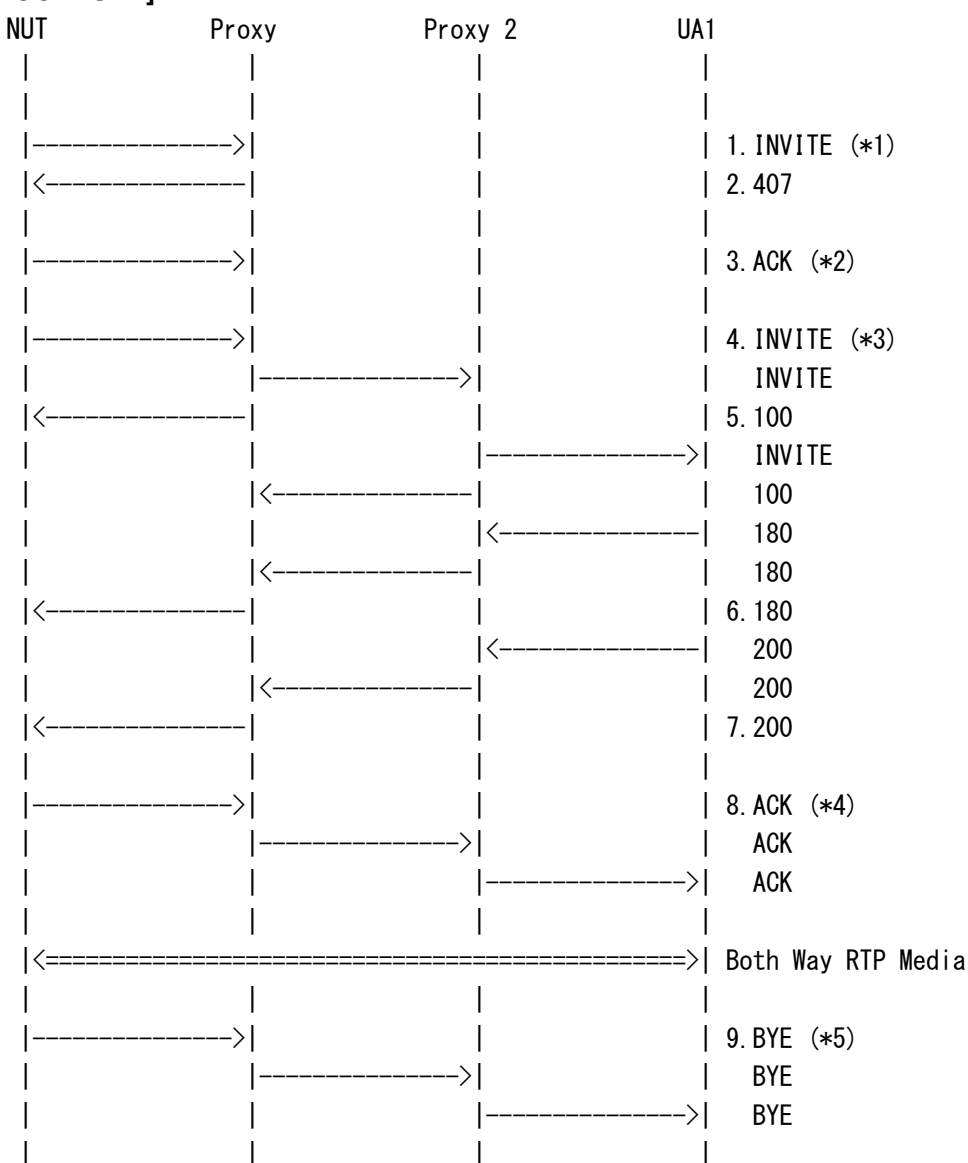
#### [INITIALIZATION]



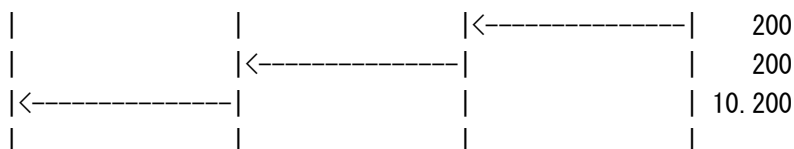


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]







1. Receive INVITE. (\*1)
2. Send 407 Proxy Authorization Required.
3. Receive ACK. (\*2)
4. Receive INVITE. (\*3)
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK. (\*4)
9. Receive BYE. (\*5)
10. Send 200 OK.

### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message  
As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*2:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:
  - See generic\_request
  - See generic\_ACK
  - See generic\_non2xx-ACK
- Header fields:
  - See generic\_request
- outside of a dialog
  - See generic\_ACK
  - See generic\_non2xx-ACK
- Bodies:
  - See generic\_ACK
  - See generic\_non2xx-ACK

\*3:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:
  - See generic\_request
  - See generic\_Initial-INVITE

- Header fields:
  - See generic\_request

- outside of a dialog
  - See generic\_Initial-INVITE

\* Call-ID

callid: Should be the same as that of "1.INVITE" request, with case-sensitivity on. [RFC3261-8-62]

\* Proxy-Authorization

Must exist. [RFC3261 22.3] ,[RFC3261-22-22]  
See generic\_digest-auth

- Bodies:



See generic\_Initial-INVITE

See generic\_SDP

\*4:ACK request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261 22.3] ,[RFC3261-22-22]

Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-9, 23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*5:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:



See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "7.200" response.  
[RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the To header field of "7.200" response. [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

#### **[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Sequences in Section 3.2

### **4.2.4 UA-2-1-4 - Session Establishment Through Two Proxies (Callee hanging up)**

#### **[NAME]**

UA-2-1-4 - Session Establishment Through Two Proxies (Callee hanging up)

#### **[PURPOSE]**

Verify that a NUT (callee) properly processes an established session on two proxies when the callee hangs up.

#### **[REQUIREMENT]**



NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

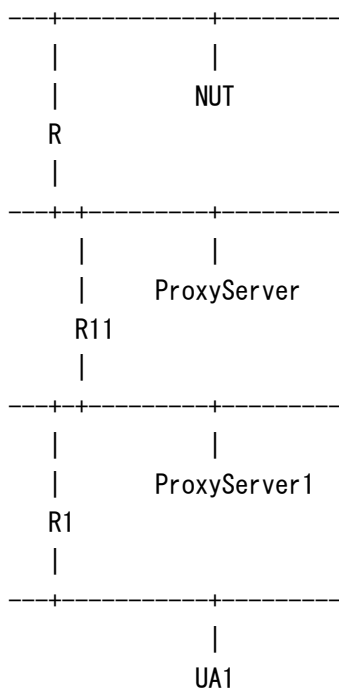
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

**[TOPOLOGY]**

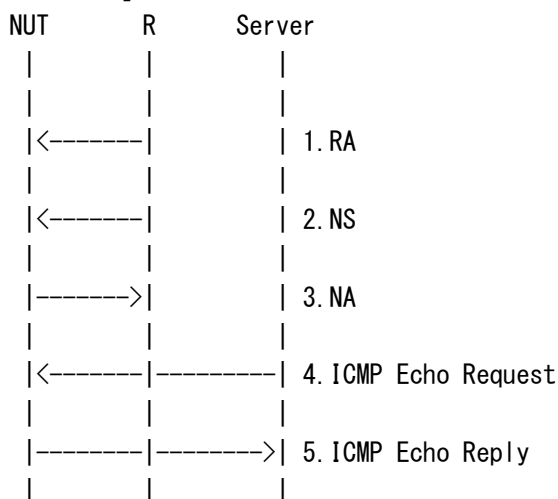


**[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

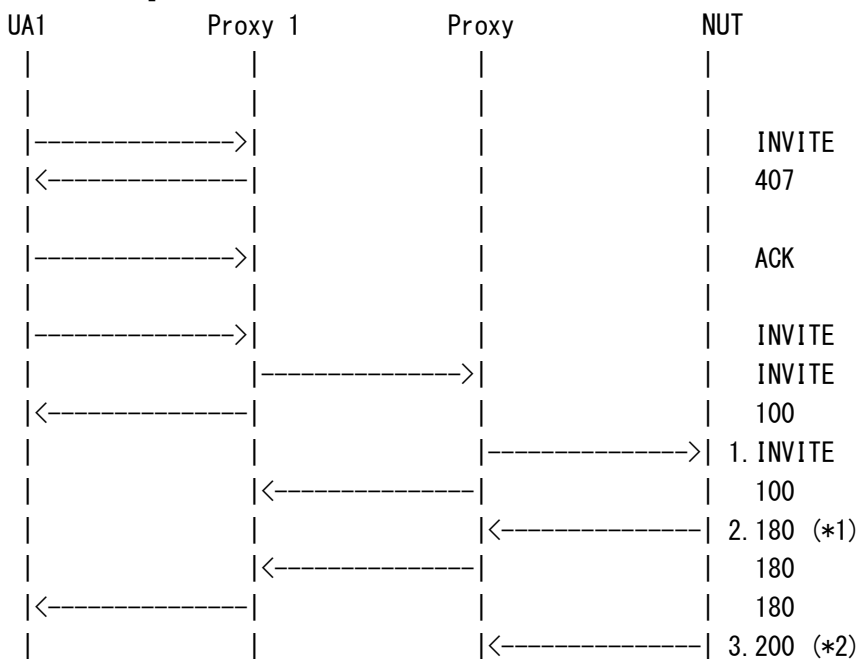
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

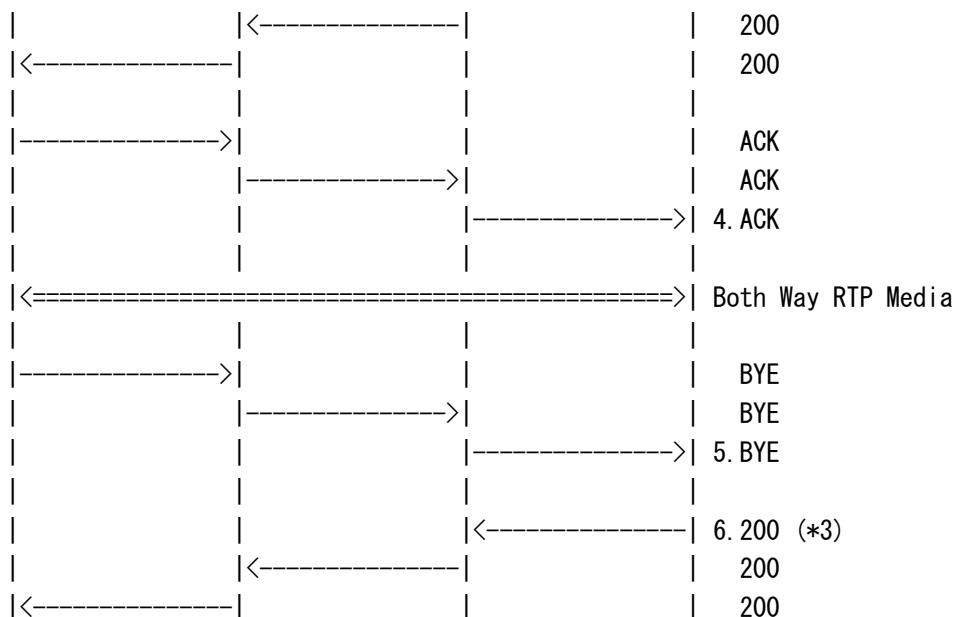
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Send BYE.
6. Receive 200 OK. (\*3)

#### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was



received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

See generic\_200-for-INVITE

Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

outside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP





\*3:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

#### **[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Sequences in Section 3.2

### **4.2.5 UA-2-1-5 - Session Establishment Through One Proxy (Caller hung up by Callee)**

#### **[NAME]**

UA-2-1-5- Session Establishment Through One Proxy (Caller hung up by Callee)

#### **[PURPOSE]**

Verify that a NUT (caller) properly processes an established session on one proxy when the callee hangs up.

#### **[REQUIREMENT]**

NONE

#### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

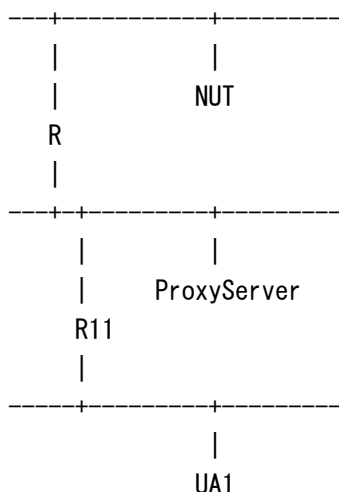
# [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@undre.test.com
UA1(Contact)	sip:UA1@client.under.test.com

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64

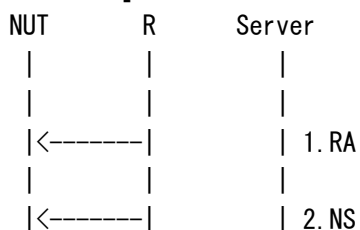
# [TOPOLOGY]

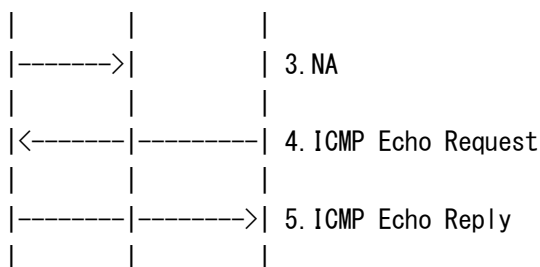


# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer/Registrar	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@under.test.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

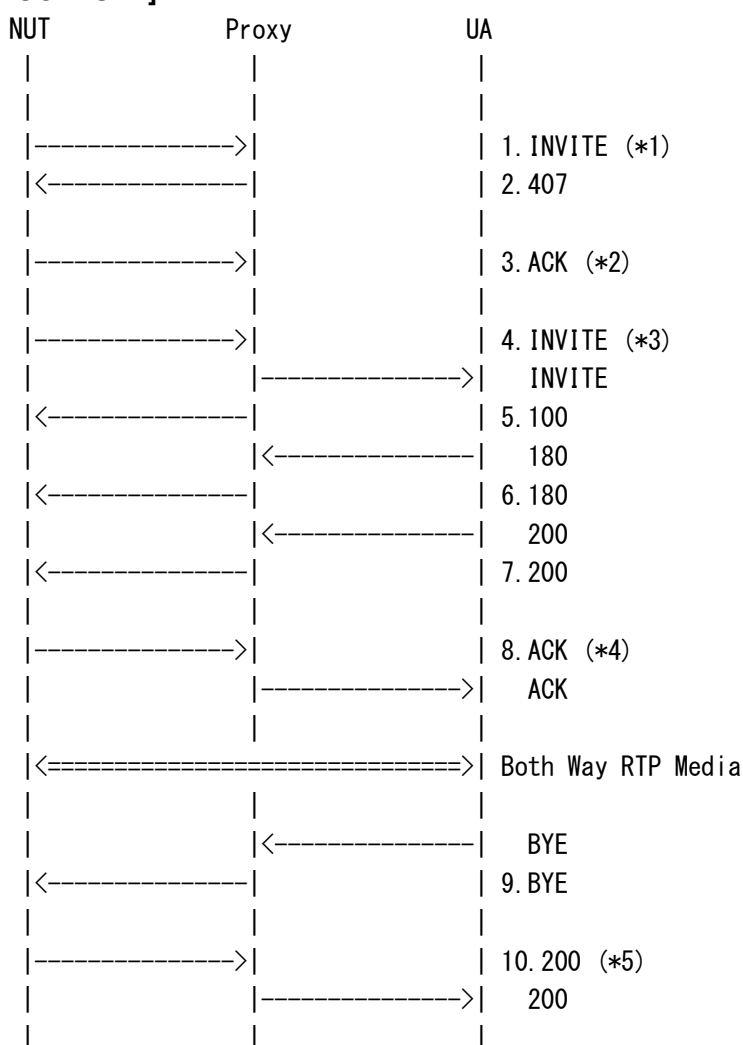
# [INITIALIZATION]





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE. (\*1)
2. Send 407 Proxy Authorization Required.
3. Receive ACK. (\*2)
4. Receive INVITE. (\*3)
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK. (\*4)
9. Send BYE.
10. Receive 200 OK. (\*5)

### **1. INVITE NUT -> Proxy**

INVITE sip:UA1@under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
/\* Proxy challenges NUT for authentication \*/

### **2. 407 Proxy Authorization Required Proxy -> NUT**

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@under.test.com



CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="f84f1cec41e6cbe5aea9c8e88d359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. ACK NUT -> Proxy

ACK sip:UA1@under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 ACK  
Content-Length: 0  
/\* NUT re-sends the INVITE with authentication  
credentials in it. \*/

### 4. INVITE NUT -> Proxy

INVITE sip:UA1@unde.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@under.test.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X



t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
/\* Proxy accepts the credentials and forwards the INVITE to Proxy  
2. Client for NUT prepares to receive data on port 49172 from the  
network. \*/

#### **5. 100 Trying Proxy -> NUT**

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### **6. 180 Ringing Proxy -> NUT**

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.under.test.com>  
Content-Length: 0

#### **7. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 147



v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 8. ACK NUT -> Proxy

ACK sip:UA1@client.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b76  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@under.test.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

## 9. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 69  
From: UA1 <sip:UA1@under.test.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

## 10. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bKnashds7



;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@under.test.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxcde76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

### **[OBSERVABLE RESULTS]**

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*2:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request





- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

\*3:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

- \* Call-ID  
callid: Should be the same as that of "1.INVITE" request, with case-sensitivity  
on. [RFC3261-8-62]

- \* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
See generic\_digest-auth

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*4:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22], [RFC3261-13-22]

Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*5:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

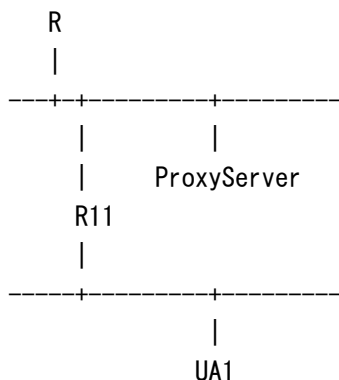
- Header fields:

See generic\_response

- inside of a dialog

\* Via

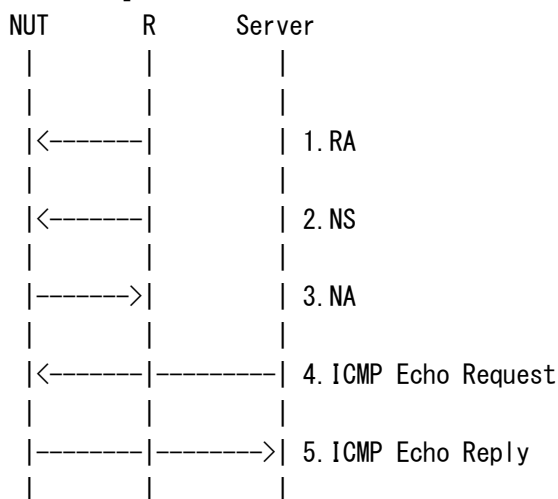




#### [CONFIGURATION for NUT]

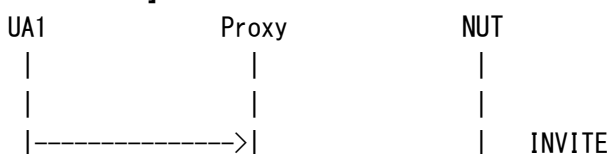
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

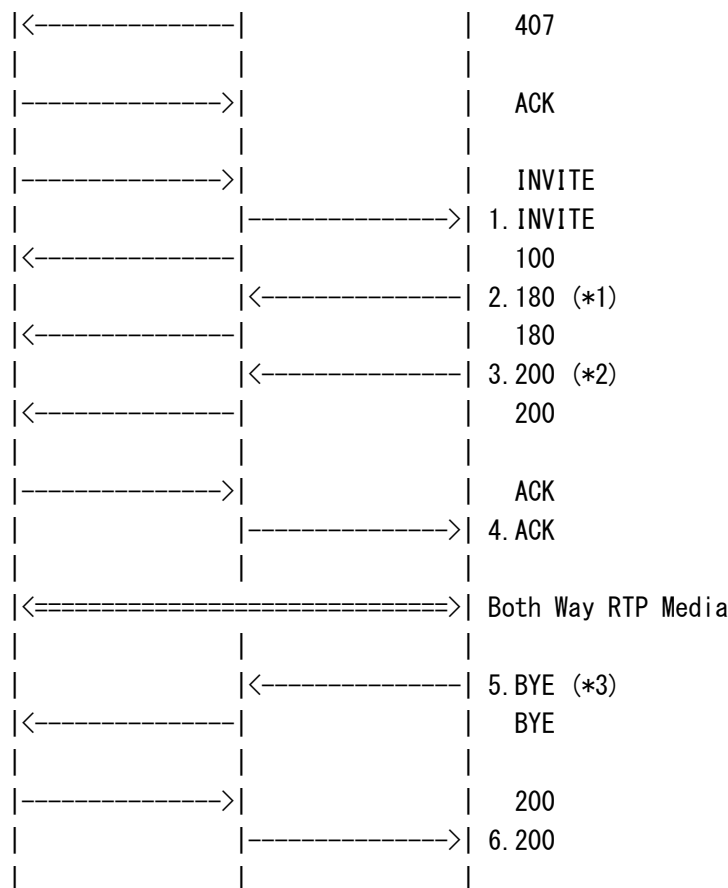
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE. (\*3)
6. Send 200 OK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
Max-Forwards: 69
Record-Route: <sip:ss.under.test.com:lr>
From: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@under.test.com
CSeq: 2 INVITE
  
```



Contact: <sip:UA1@client.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 2. 180 Ringing NUT -> Proxy

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>  
From: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Length: 0

## 3. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>  
From: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp



Content-Length: 147

v=0  
o=NUT 2890844527 2890844527 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### 4. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP client.under.test.com:5060;branch=z9hG4bK74b76  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 69  
From: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

#### 5. BYE NUT -> Proxy

BYE sip:UA1@client.under.test.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
Max-Forwards: 70  
Route: <sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

#### 6. 200 OK Proxy -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
Content-Length: 0



## [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\*2:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response





- outside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

\*3:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request. [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-35]

\* From



tag-param: Must equal that contained in the To header field of "3.200" response.  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

**[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Sequences in Section 3.2

## 4.2.7 UA-2-1-7 - Unsuccessful No Answer [CANCEL] (Caller)

**[NAME]**

UA-2-1-7 - Unsuccessful No Answer [CANCEL] (Caller)

**[PURPOSE]**

Verify that a NUT (Caller) properly sends a CANCEL request when the UA sends and receives no response.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

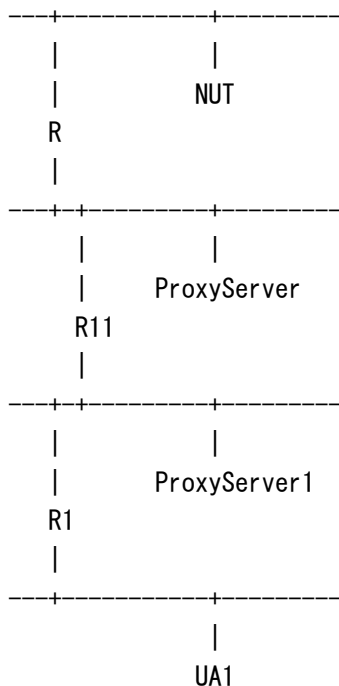
**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64



ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

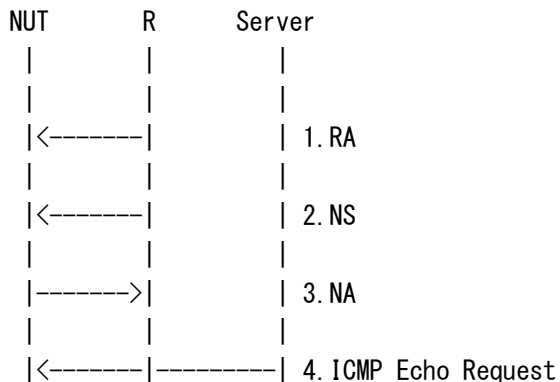
#### [TOPOLOGY]

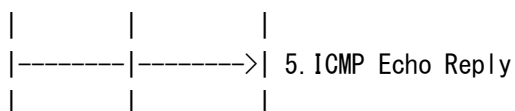


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

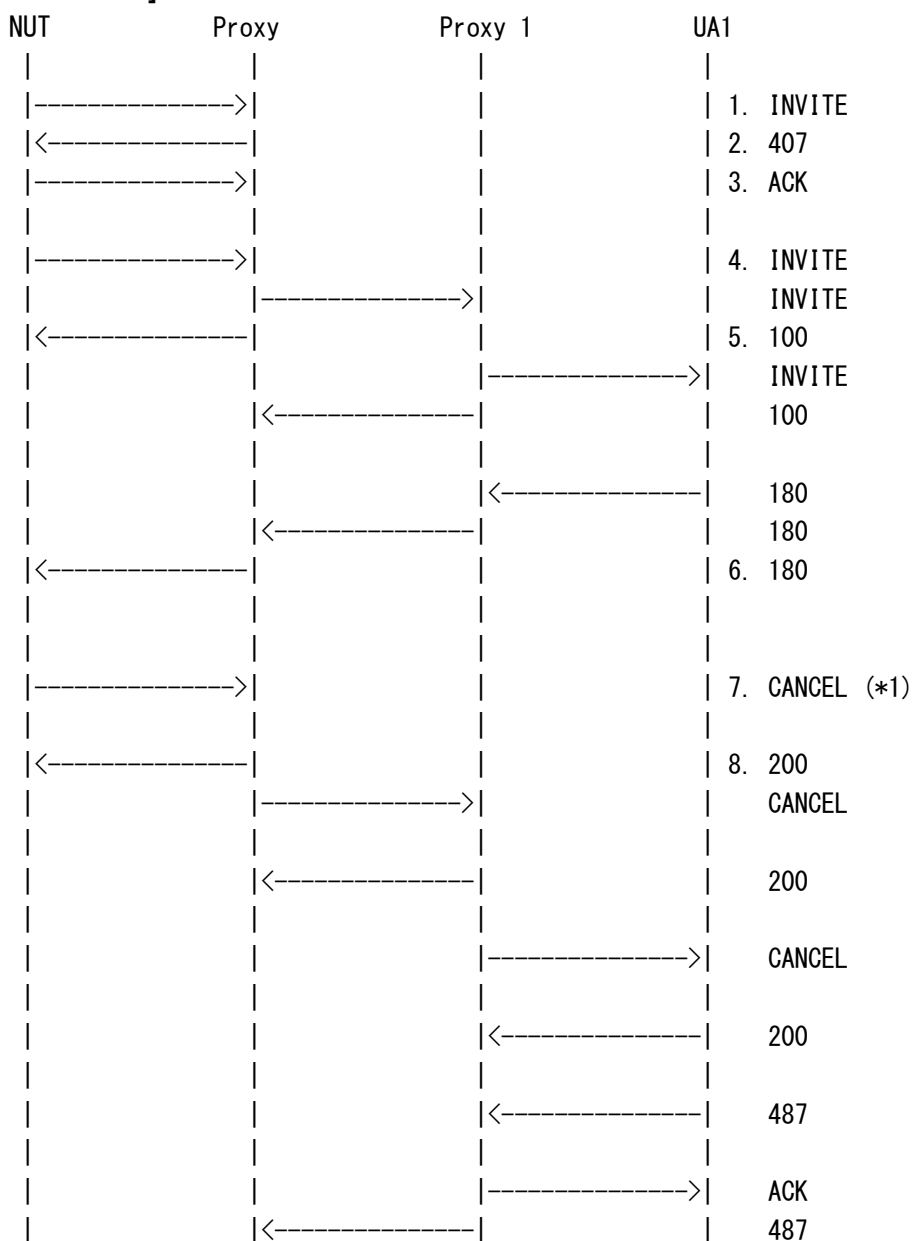
#### [INITIALIZATION]

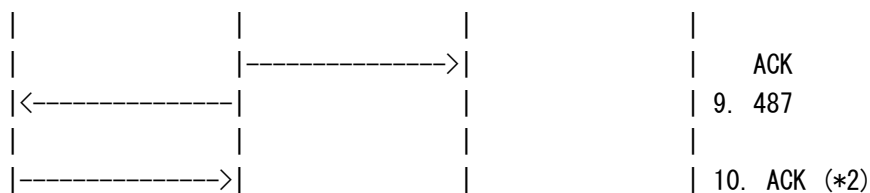




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Receive CANCEL. (\*1)
8. Send 200 OK.
9. Send 487 Request Terminated.
10. Receive ACK. (\*2)

**=== Message example ===**

**1. INVITE NUT -> Proxy**

```
INVITE sip:UA1@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43
Route: <sip:ss.under.test.com;lr>
Max-Forwards: 70
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 2xTb9vxSit55XU7p8@under.test.com
CSeq: 1 INVITE
Contact: <sip:NUT@node.under.test.com>
Supported:
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Type: application/sdp
Content-Length: 151
```

```
v=0
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X
s=-
c=IN IP6 3ffe:501:ffff:5::X
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

**2. 407 Proxy Authorization Required Proxy -> NUT**

```
SIP/2.0 407 Proxy Authorization Required
```



Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="f84f1cec41e6cbe5aea9c8e88d359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 ACK  
Content-Length: 0

### 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0



o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### **5. 100 Trying Proxy -> NUT**

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### **6. 180 Ringing Proxy -> NUT**

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Length: 0

#### **7. CANCEL NUT -> Proxy**

CANCEL sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 CANCEL  
Content-Length: 0

#### **8. 200 OK Proxy -> NUT**

SIP/2.0 200 OK



Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 CANCEL  
Content-Length: 0

#### **9. 487 Request Terminated Proxy -> NUT**

SIP/2.0 487 Request Terminated  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### **10. ACK NUT -> Proxy**

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

#### **[OBSERVABLE RESULTS]**

\*1: CANCEL request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request





See generic\_CANCEL

- Header fields:

See generic\_request

- outside of a dialog

See generic\_CANCEL

\* Route

Must exist. [RFC3261-12-48]

route-param: Must include Route header field's values of the request being cancelled. [RFC3261-9-5]

- Bodies:

See generic\_CANCEL

\*2:ACK request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_non2xx-ACK

- Header fields:

See generic\_request

- outside of a dialog

See generic\_ACK

See generic\_non2xx-ACK

- Bodies:

See generic\_ACK

See generic\_non2xx-ACK

## [REFERENCE]

RFC3665 SIP Basic Call Flow Examples

See Sequence in Section 3.8

## 4.2.8 UA-2-1-8 - Unsuccessful No Answer [CANCEL] (Callee)

### [NAME]

UA-2-1-8 - Unsuccessful No Answer [CANCEL] (Callee)

### [PURPOSE]

Verify that a NUT (Callee) properly processes when the UA doesn't send any response and receives a CANCEL request.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

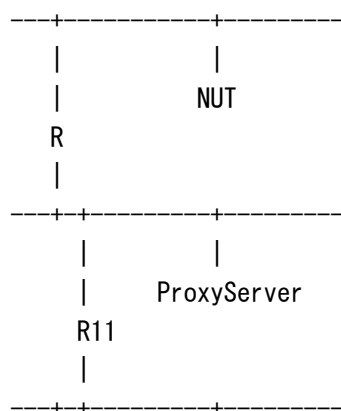
### [PARAMETER]

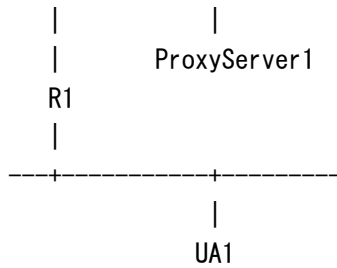
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]

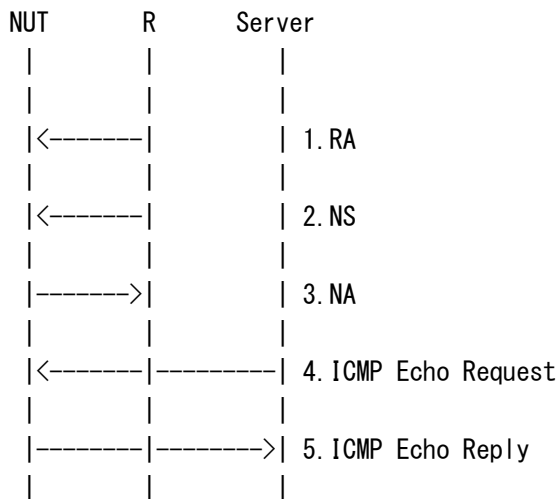




#### [CONFIGURATION for NUT]

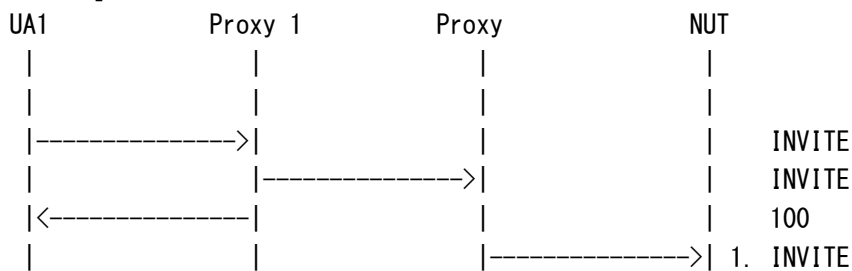
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

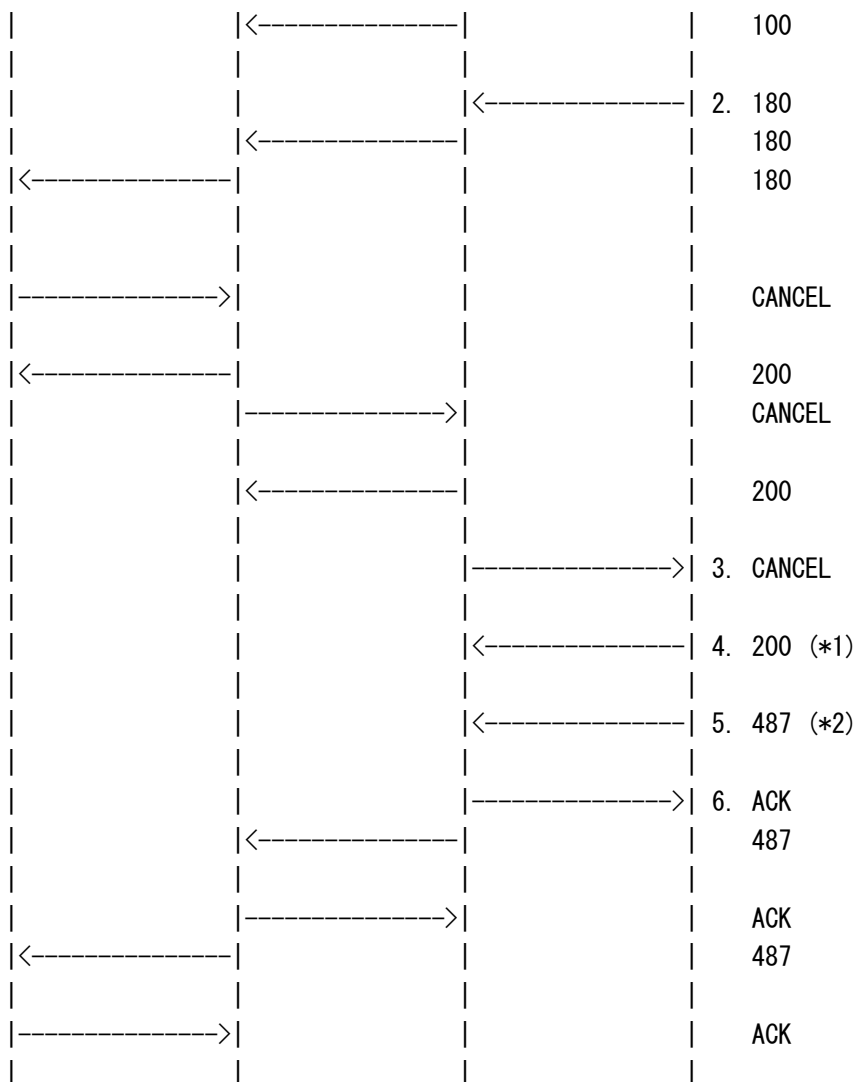
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Send CANCEL.
4. Receive 200 OK. (\*1)
5. Receive 487 Request Terminated. (\*2)
6. Send ACK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
  
```



;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
    <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## **2. 180 Ringing NUT -> Proxy**

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
    ;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
    ;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
    ;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
    <sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Length: 0

## **3. CANCEL Proxy -> NUT**

CANCEL sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Max-Forwards: 70



From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 CANCEL  
Content-Length: 0

#### 4. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 CANCEL  
Content-Length: 0

#### 5. 487 Request Terminated NUT -> Proxy

SIP/2.0 487 Request Terminated  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Content-Length: 0

#### 6. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Max-Forwards: 70  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 ACK  
Content-Length: 0

### [OBSERVABLE RESULTS]

- \*1:200 response from NUT.  
As a SIP Message,



See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* To

tag-param: Should be the same as the To tag in the response to the original request. [RFC3261-9-16]

- Bodies:

See generic\_response

\*2:487 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "487".

- Header fields:

See generic\_response

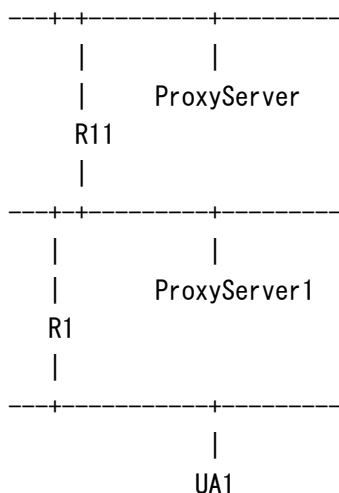
\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]



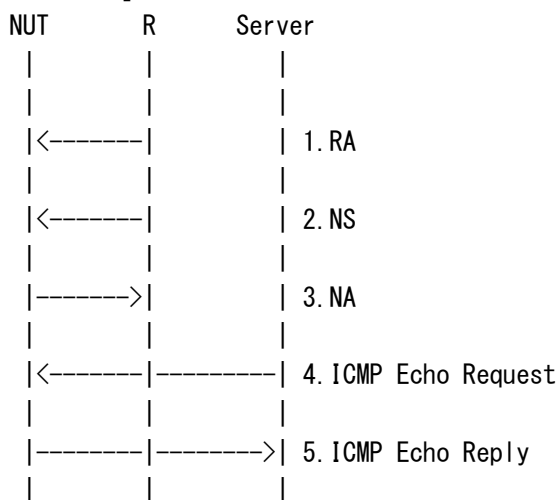




#### [CONFIGURATION for NUT]

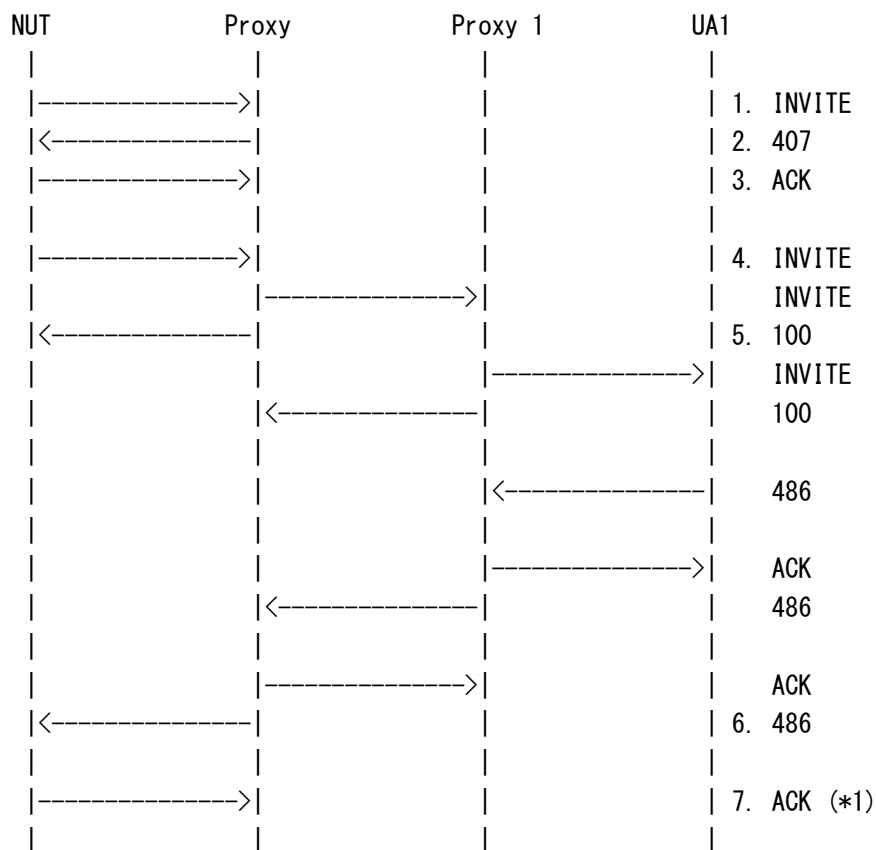
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 486 Busy Here.
7. Receive ACK. (\*1)

### === Message example ===

#### 1. INVITE NUT -> Proxy

```

INVITE sip:UA1@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43
Route: <sip:ss.under.test.com;lr>
Max-Forwards: 70
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 2xTb9vxSit55XU7p8@under.test.com
CSeq: 1 INVITE
Contact: <sip:NUT@node.under.test.com>
Supported:

```



Allow: INVITE, ACK, CANCEL, OPTIONS, BYE

Content-Type: application/sdp

Content-Length: 151

v=0

o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X

s=-

c=IN IP6 3ffe:501:ffff:5::X

t=0 0

m=audio 49172 RTP/AVP 0

a=rtpmap:0 PCMU/8000

## 2. 407 Proxy Authorization Required Proxy -> NUT

SIP/2.0 407 Proxy Authorization Required

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43

;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf

Call-ID: 2xTb9vxSit55XU7p8@under.test.com

CSeq: 1 INVITE

Proxy-Authenticate: Digest realm="under.test.com", qop="auth",

nonce="dc3a5ab2530aa93112cf5904ba7d88fa",

opaque="", stale=FALSE, algorithm=MD5

Content-Length: 0

## 3. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43

Max-Forwards: 70

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf

Call-ID: 3848276298220188511@under.test.com

CSeq: 1 ACK

Content-Length: 0

## 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9

Route: <sip:ss.under.test.com;lr>

Max-Forwards: 70

Proxy-Authorization: Digest username="NUT",

realm="under.test.com",

nonce="dc3a5ab2530aa93112cf5904ba7d88fa", opaque="",



qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151  
  
v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
/\*Client for NUT prepares to receive data on port 49172 from the  
network.\*/

#### **5. 100 Trying Proxy -> NUT**

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### **6. 486 Busy Here Proxy -> NUT**

SIP/2.0 486 Busy Here  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### **7. ACK NUT -> Proxy**



ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="dc3a5ab2530aa93112cf5904ba7d88fa", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

#### **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

#### **[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Sequence in Section 3.9



#### 4.2.10 UA-2-2-2 - Unsuccessful Busy (Callee)

##### [NAME]

UA-2-2-2 - Unsuccessful Busy (Callee)

##### [PURPOSE]

Verify that a NUT (callee) properly executes an unsuccessful processing when the caller is busy.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

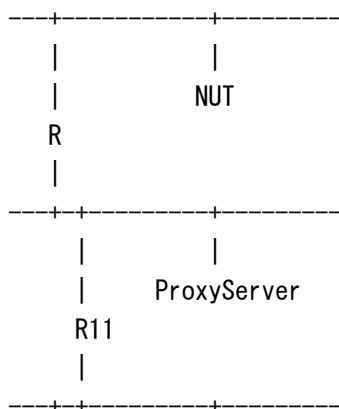
##### [PARAMETER]

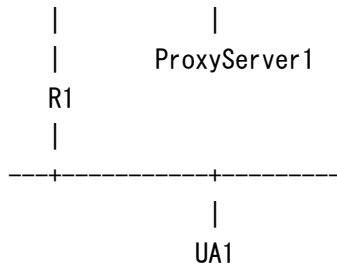
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

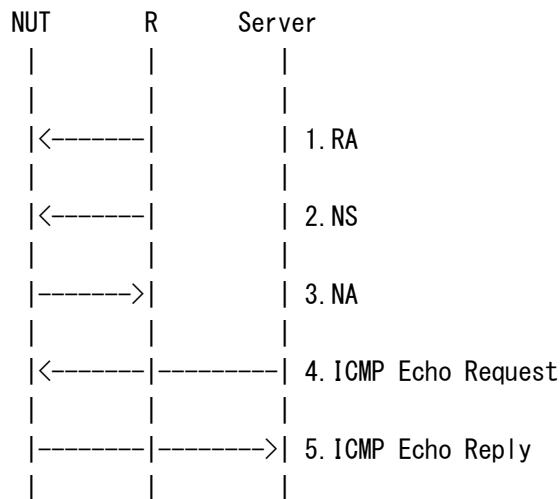




#### [CONFIGURATION for NUT]

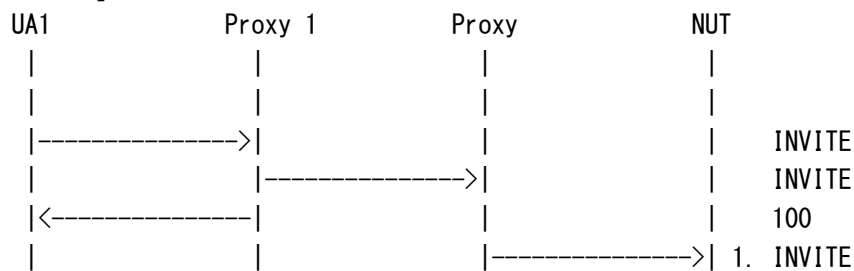
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

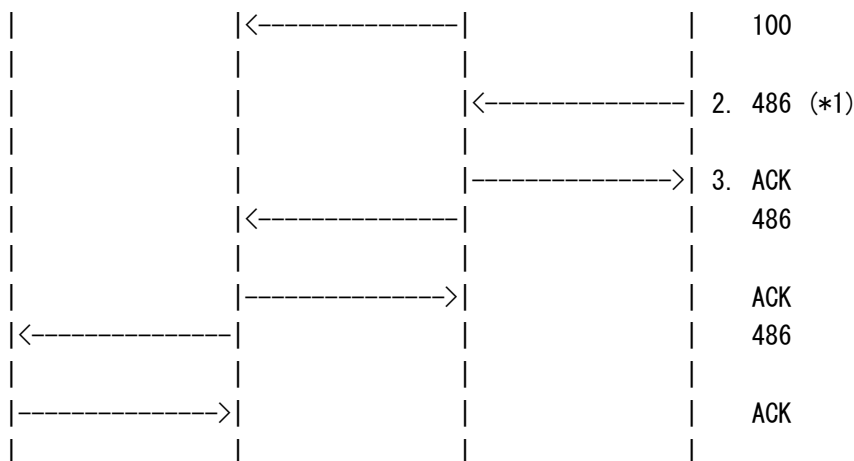
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 486 Busy Here. (\*1)
3. Send ACK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Supported:
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Type: application/sdp
Content-Length: 151

v=0
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0

```





m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 2. 486 Busy Here NUT -> Proxy

SIP/2.0 486 Busy Here  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Content-Length: 0

## 3. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Max-Forwards: 70  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 ACK  
Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:486 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

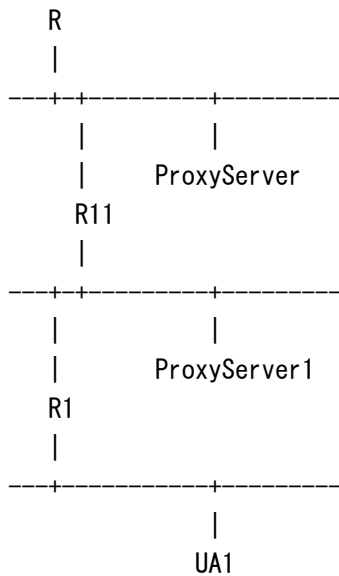
- Status-Line:  
See generic\_response  
Status-Code: Must be "486". [RFC3261-13-32]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

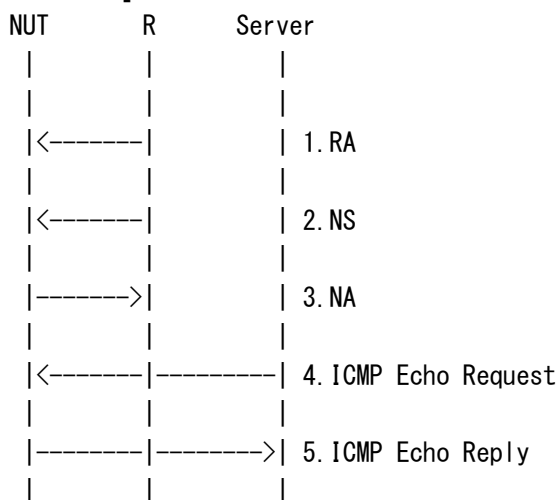




#### [CONFIGURATION for NUT]

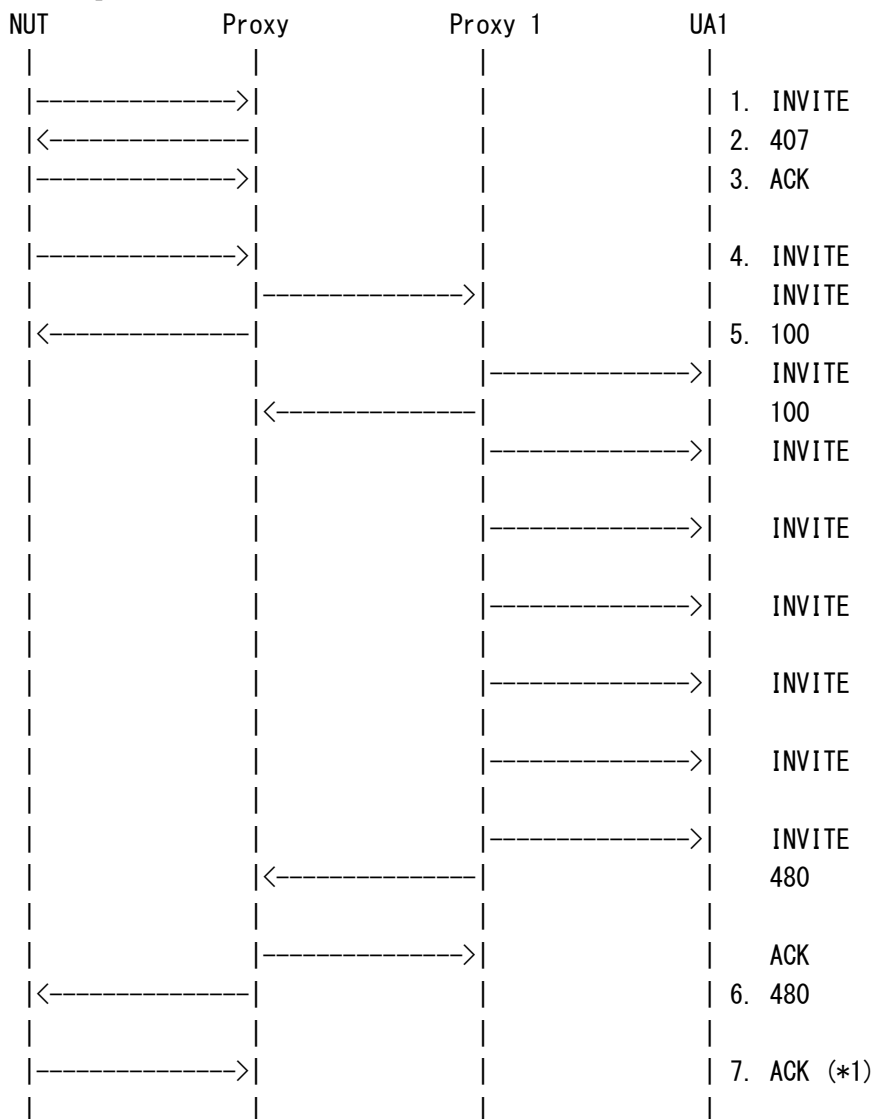
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]



1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 480 No Response.
7. Receive ACK. (\*1)

### === Message example ===

#### 1. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0



Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## **2. 407 Proxy Authorization Required Proxy -> NUT**

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="cf5904ba7d8dc3a5ab2530aa931128fa",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

## **3. ACK NUT -> Proxy**

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 ACK  
Content-Length: 0



#### 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
    realm="under.test.com",  
    nonce="cf5904ba7d8dc3a5ab2530aa931128fa", opaque="",  
    qop=auth, nc=00000004, cnonce="6f54a149",  
    uri="sip:UA1@atlanta.example.com",  
    response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### 5. 100 Trying Proxy -> NUT

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
    ;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### 6. 480 Temporarily Unavailable Proxy -> NUT

SIP/2.0 480 Temporarily Unavailable  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
    ;received=3ffe:501:ffff:5::X



From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

## 7. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="cf5904ba7d8dc3a5ab2530aa931128fa", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

## [OBSERVABLE RESULTS]

\*1:ACK request from NUT.

- As a SIP Message,
  - See generic\_message
- As a SIP request,
  - Request-Line:
    - See generic\_request
    - See generic\_ACK
    - See generic\_non2xx-ACK
  - Header fields:
    - See generic\_request
  - outside of a dialog
    - See generic\_ACK
    - See generic\_non2xx-ACK
  - Bodies:
    - See generic\_ACK
    - See generic\_non2xx-ACK



#### [REFERENCE]

RFC3665 SIP Basic Call Flow Examples

See Sequence in Section 3.10

#### [RFC3261-21-9]

##### 21.4 Request Failure 4xx

4xx responses are definite failure responses from a particular server. The client **SHOULD NOT** retry the same request without modification (for example, adding appropriate authorization). However, the same request to a different server might be successful.

### 4.2.12 UA-2-2-4 - Unsuccessful Temporarily Unavailable (Caller)

#### [NAME]

UA-2-2-4 - Unsuccessful Temporarily Unavailable (Caller)

#### [PURPOSE]

Verify that a NUT (caller) properly executes an unsuccessful processing by 480 (Temporarily Unavailable.)

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

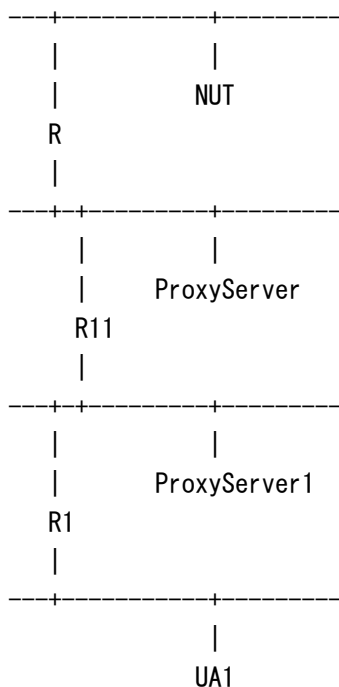
#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64



ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64
--------------------	-------------------------

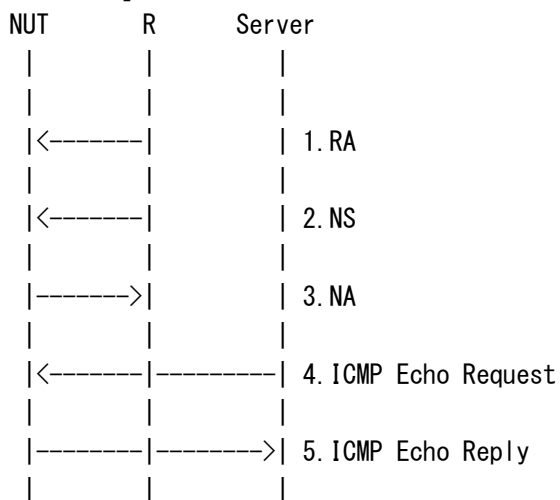
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

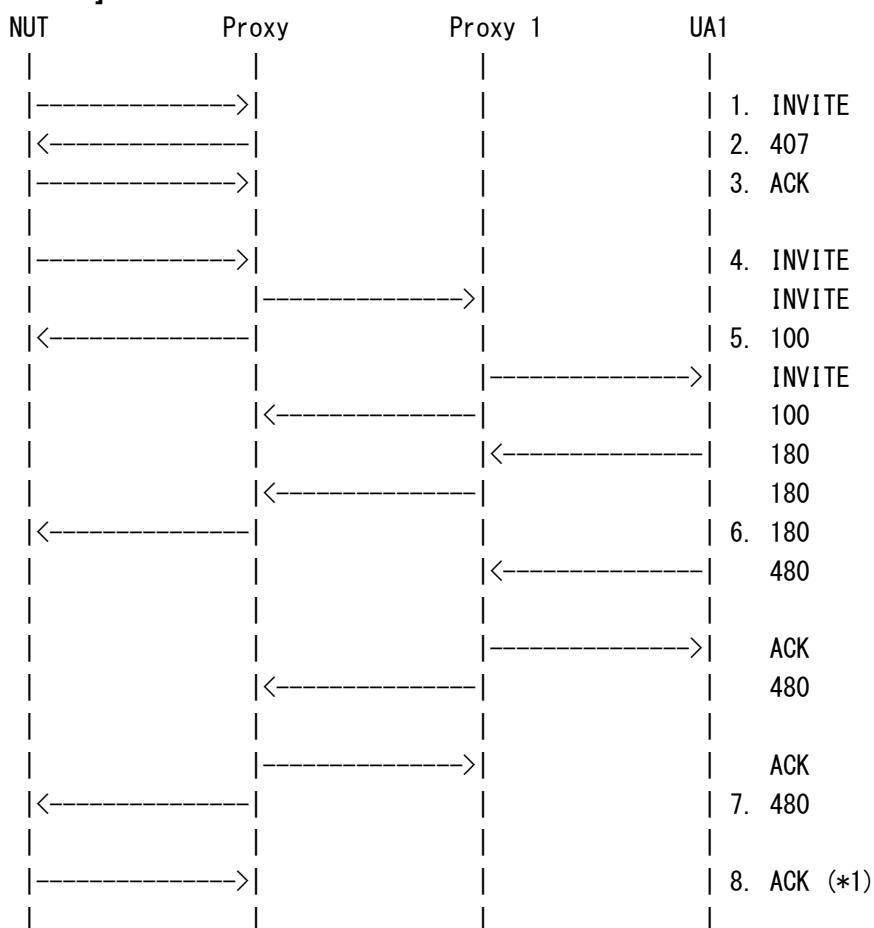
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 480 Temporarily Unavailable.
8. Receive ACK. (\*1)

=== Message example ===



### 1. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

### 2. 407 Proxy Authorization Required Proxy -> NUT

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="aa9311cf5904ba7d8dc3a5ab253028fa",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

### 3. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com



CSeq: 1 ACK  
Content-Length: 0

#### 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Route: <sip:ss.under.test.com;lr>  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="aa9311cf5904ba7d8dc3a5ab253028fa", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Supported:  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
/\*Client for NUT prepares to receive data on port 49172 from the  
network.\*/

#### 5. 100 Trying Proxy -> NUT

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0



## 6. 180 Ringing Proxy -> NUT

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X

Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@under.test.com

CSeq: 2 INVITE

Contact: <sip:UA1@client.atlanta.example.com>

Content-Length: 0

## 7. 480 Temporarily Unavailable Proxy -> NUT

SIP/2.0 480 Temporarily Unavailable

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@under.test.com

CSeq: 2 INVITE

Content-Length: 0

## 8. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9

Max-Forwards: 70

Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="aa9311cf5904ba7d8dc3a5ab253028fa", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@under.test.com

CSeq: 2 ACK

Content-Length: 0

## [OBSERVABLE RESULTS]

\*1:ACK request from NUT.

As a SIP Message,

See generic\_message



As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_non2xx-ACK

- Header fields:

See generic\_request

- outside of a dialog

See generic\_ACK

See generic\_non2xx-ACK

- Bodies:

See generic\_ACK

See generic\_non2xx-ACK

#### **[REFERENCE]**

RFC3665 SIP Basic Call Flow Examples

See Sequence in Section 3.11

[RFC3261-21-9]

21.4 Request Failure 4xx

4xx responses are definite failure responses from a particular server. The client SHOULD NOT retry the same request without modification (for example, adding appropriate authorization). However, the same request to a different server might be successful.

## **4.3 Transaction**

### **4.3.1 UA-4-1-1 - INVITE Client Transaction (Retransmission)**

#### **[NAME]**

UA-4-1-1 - INVITE Client Transaction (retransmission)

#### **[PURPOSE]**

Verify that a NUT properly retransmits an INVITE client transaction

#### **[REQUIREMENT]**

NONE

# [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

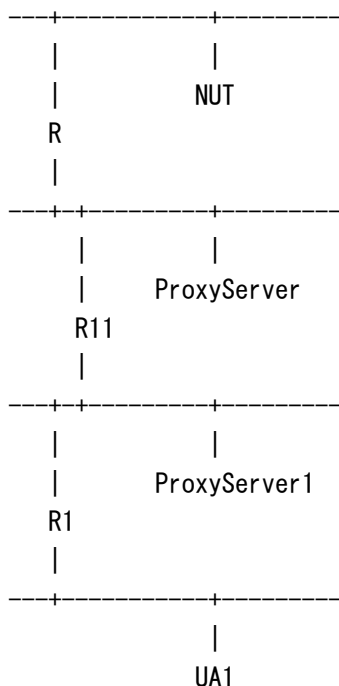
# [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# [TOPOLOGY]

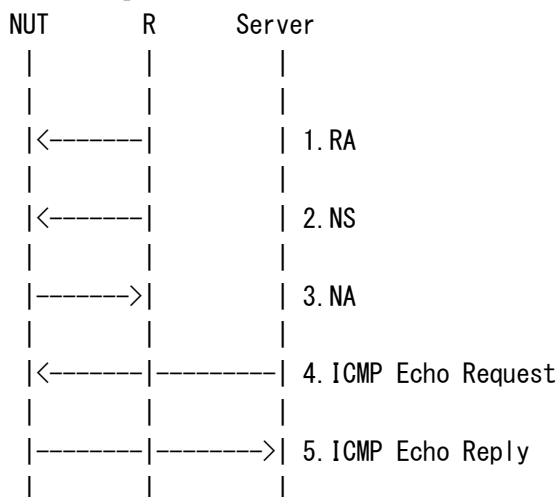


# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

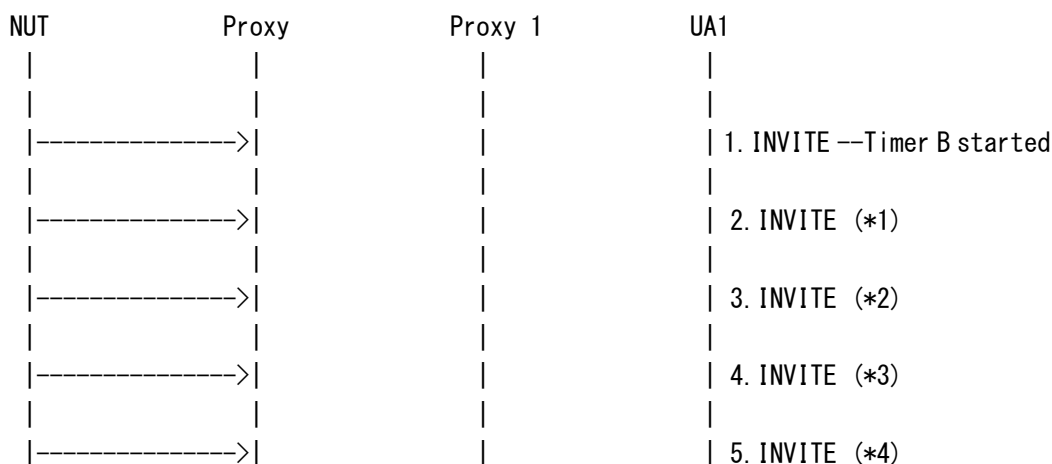


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

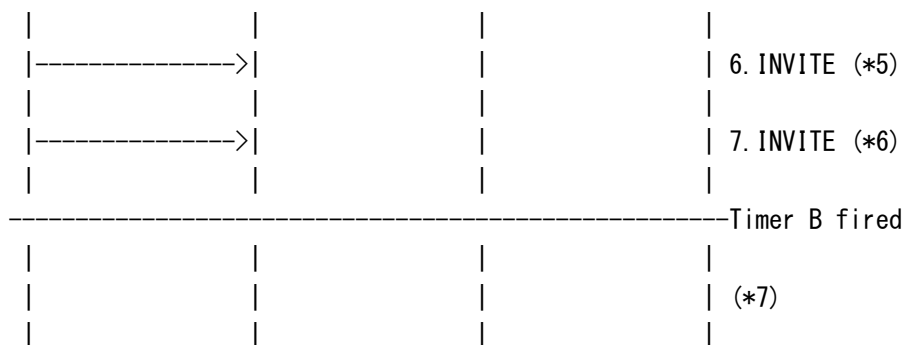
#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.







1. Receive INVITE.
2. Receive INVITE. (\*1)
3. Receive INVITE. (\*2)
4. Receive INVITE. (\*3)
5. Receive INVITE. (\*4)
6. Receive INVITE. (\*5)
7. Receive INVITE. (\*6)
- (\*7)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:INVITE request from NUT.

Must be retransmitted after Timer A(= T1 sec.) fired.  
[RFC3261-17-4][RFC3261-17-7]

Recommended not to be retransmitted with intervals that is shorter than 500msec.  
[RFC3261-17-12]

\*2:INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission(2\*T1).  
[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14]

\*3:INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission(2\*T1).  
[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14]



\*4:INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission( $2 \cdot T1$ ).  
[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14]

\*5:INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission( $2 \cdot T1$ ).  
[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14]

\*6:INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission( $2 \cdot T1$ ).  
[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14]

\*7:INVITE request from NUT.

INVITE request Must not be retransmitted after Timer  
B( $64 \cdot t1$ ). [RFC3261-17-6][RFC3261-17-11]  
Must not send ACK request. [RFC3261-17-16]

## [REFERENCE]

[RFC3261-12-56, 57]

### 12.2.1.2 Processing the Responses

The UAC will receive responses to the request from the transaction layer. If the client transaction returns a timeout, this is treated as a 408 (Request Timeout) response.

If the response for a request within a dialog is a 481 (Call/Transaction Does Not Exist) or a 408 (Request Timeout), the UAC SHOULD terminate the dialog. A UAC SHOULD also terminate a dialog if no response at all is received for the request (the client transaction would inform the TU about the timeout.)

For INVITE initiated dialogs, terminating the dialog consists of sending a BYE.

[RFC3261-17-4, 6, 7, 8, 9, 10, 11, 14, 15, 16]

### 17.1.1.2 Formal Description



\*snip\* If an unreliable transport is being used, the client transaction **MUST** start timer A with a value of  $T1$ .  
\*snip\* For any transport, the client transaction **MUST** start timer B with a value of  $64 \cdot T1$  seconds (Timer B controls transaction timeouts).

When timer A fires, the client transaction **MUST** retransmit the request by passing it to the transport layer, and **MUST** reset the timer with a value of  $2 \cdot T1$ . The formal definition of retransmit within the context of the transaction layer is to take the message previously sent to the transport layer and pass it to the transport layer once more.

When timer A fires  $2 \cdot T1$  seconds later, the request **MUST** be retransmitted again (assuming the client transaction is still in this state). This process **MUST** continue so that the request is retransmitted with intervals that double after each transmission. These retransmissions **SHOULD** only be done while the client transaction is in the "calling" state.

\*snip\*  
Whatever the value of  $T1$ , the exponential backoffs on retransmissions described in this section **MUST** be used.

If the client transaction is still in the "Calling" state when timer B fires, the client transaction **SHOULD** inform the TU that a timeout has occurred. The client transaction **MUST NOT** generate an ACK. The value of  $64 \cdot T1$  is equal to the amount of time required to send seven requests in the case of an unreliable transport.

#### **4.3.2 UA-4-1-2 - INVITE Client Transaction (Receipt of 180 Ringing and Stop of retransmission)**

##### **[NAME]**

UA-4-1-2 - INVITE Client Transaction (Receipt of 180 Ringing and Stop of retransmission)

##### **[PURPOSE]**

Verify that a NUT properly stops retransmitting another INVITE request when receiving a 180 response.

##### **[REQUIREMENT]**

NONE

# **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

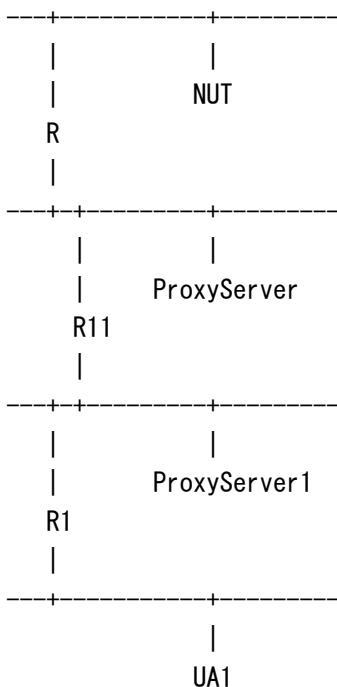
# **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

# **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# **[TOPOLOGY]**

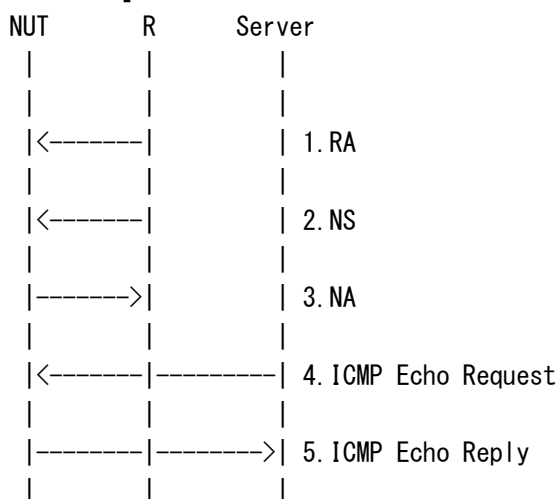


# **[CONFIGURATION for NUT]**



NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

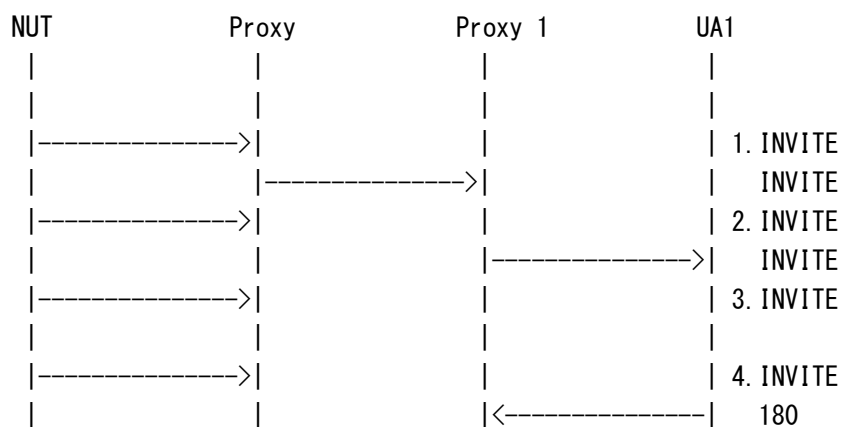


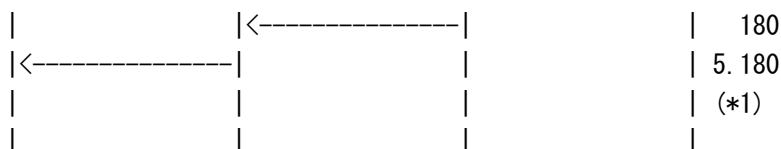
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Receive INVITE.
  2. Receive INVITE.
  3. Receive INVITE.
  4. Receive INVITE.
  5. Send 180 Ringing.
- (\*1)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:after 180 response from Proxy.

Should not retransmit INVITE request. [RFC3261-17-11][RFC3261-17-17]

#### [REFERENCE]

[RFC3261-17-11, 17]

17.1.1.2 Formal Description

\*snip\*

These retransmissions **SHOULD** only be done while the client transaction is in the "calling" state.

If the client transaction receives a provisional response while in the "Calling" state, it transitions to the "Proceeding" state. In the "Proceeding" state, the client transaction **SHOULD NOT** retransmit the request any longer. \*snip\*

### 4.3.3 UA-4-1-3 - INVITE Client Transaction (Stop of ACK upon Timer D fired)

#### [NAME]

UA-4-1-3 - INVITE Client Transaction (Stop of ACK upon Timer D fired)

#### [PURPOSE]

Verify that a NUT properly processes INVITE client transaction of stopping sending ACK when Timer D fires.

**[REQUIREMENT]**

NONE

**[TARGET]**

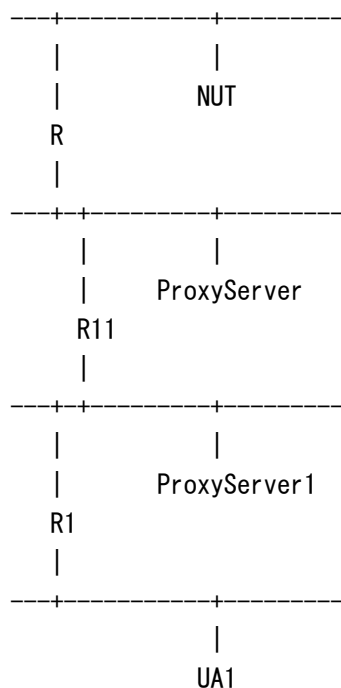
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

**[ADDRESS]**

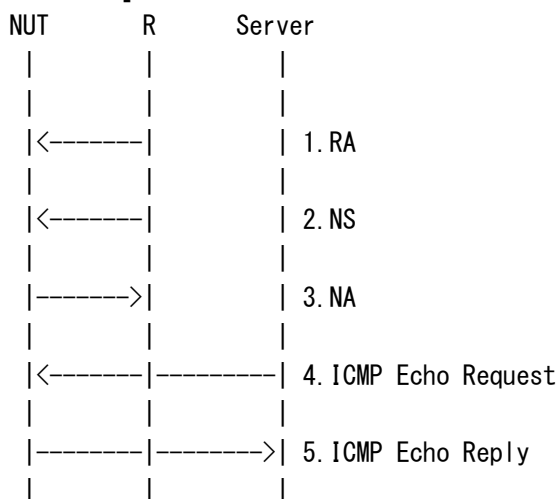
NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

**[TOPOLOGY]**

# **[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

# **[INITIALIZATION]**

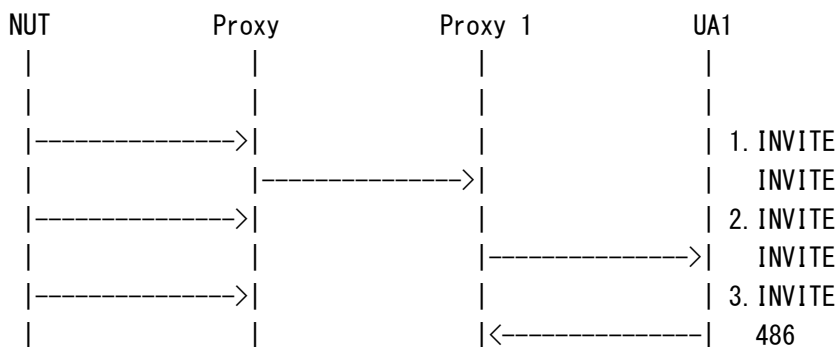


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

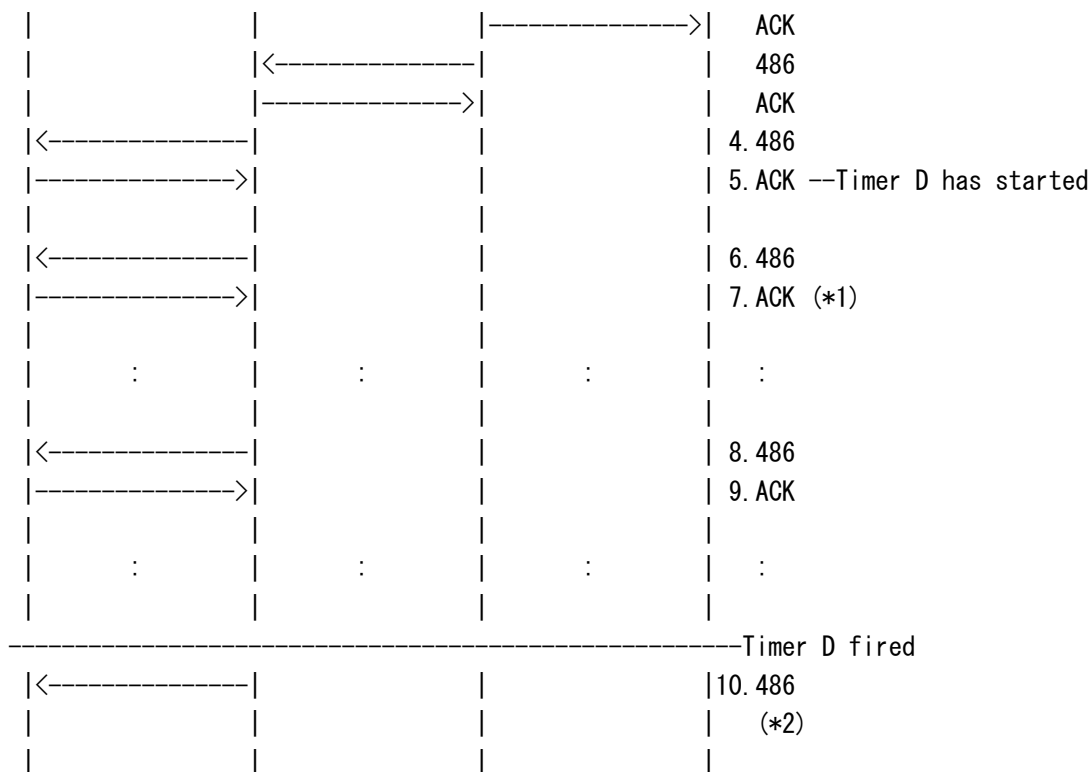
# **[PROCEDURE]**

Note: This sequence is an example.

In some cases, the number of retransmission changes.







- ## [OBSERVABLE RESULTS]

\*\* response(486 is used below) may be 300-699 response.

ACK request Must be retransmitted before Timer D( $\geq 32$ sec.) fires.  
[RFC3261-17-24][RFC3261-17-25]



\*2:after 486 response(Timer D is already fired) from Proxy.

ACK request Must not be retransmitted. [RFC3261-17-27]

#### [REFERENCE]

[RFC3261-17-20, 24, 28, 27]

##### 17.1.1.2 Formal Description

When in either the "Calling" or "Proceeding" states, reception of a response with status code from 300-699 MUST cause the client transaction to transition to "Completed". \*snip\* The client transaction SHOULD start timer D when it enters the "Completed" state, with a value of at least 32 seconds for unreliable transports, and a value of zero seconds for reliable transports. \*snip\*

When in either the "Calling" or "Proceeding" states, reception of a 2xx response MUST cause the client transaction to enter the "Terminated" state, and \*snip\*

If timer D fires while the client transaction is in the "Completed" state, the client transaction MUST move to the terminated state.

#### **4.3.4 UA-4-1-4 - Non-INVITE Client Transaction (Stop of retransmission of CANCEL upon Timer F fired)**

##### [NAME]

UA-4-1-4 - Non-INVITE Client Transaction (Stop of retransmission of CANCEL upon Timer F fired)

##### [PURPOSE]

Verify that a NUT properly processes non-INVITE client transaction of stopping retransmitting another CANCEL request when Timer F fires.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

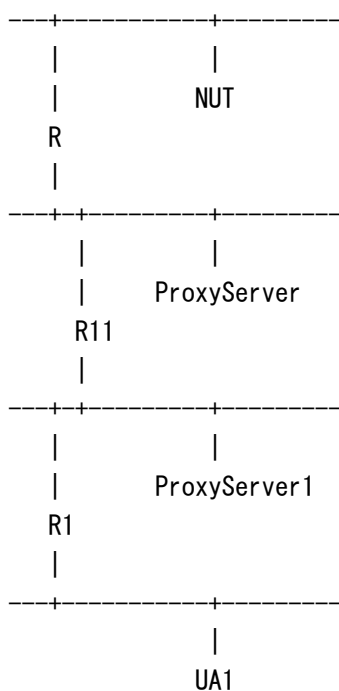
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr

UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

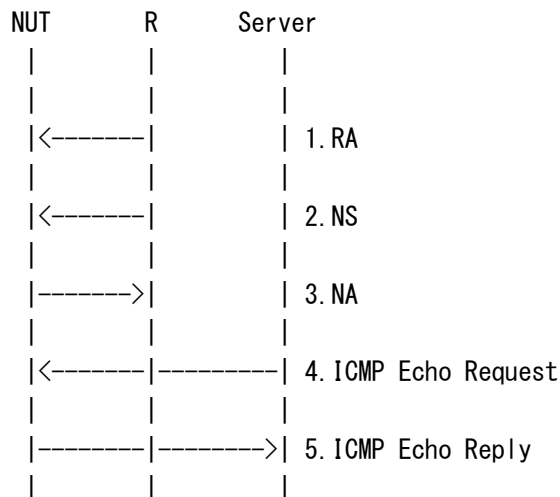
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

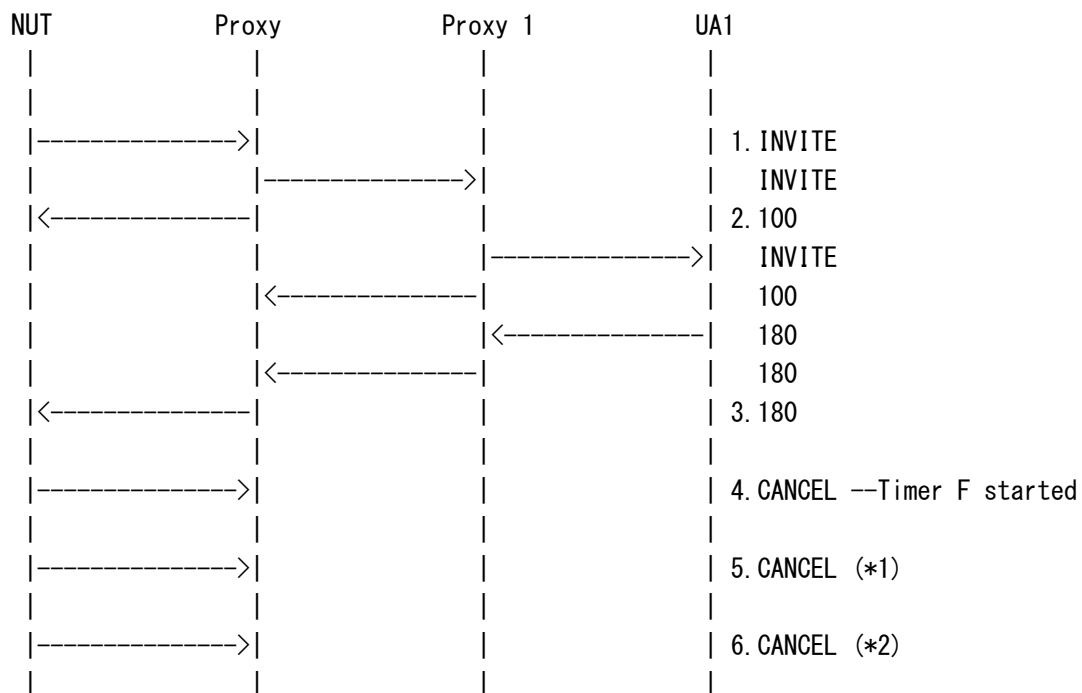


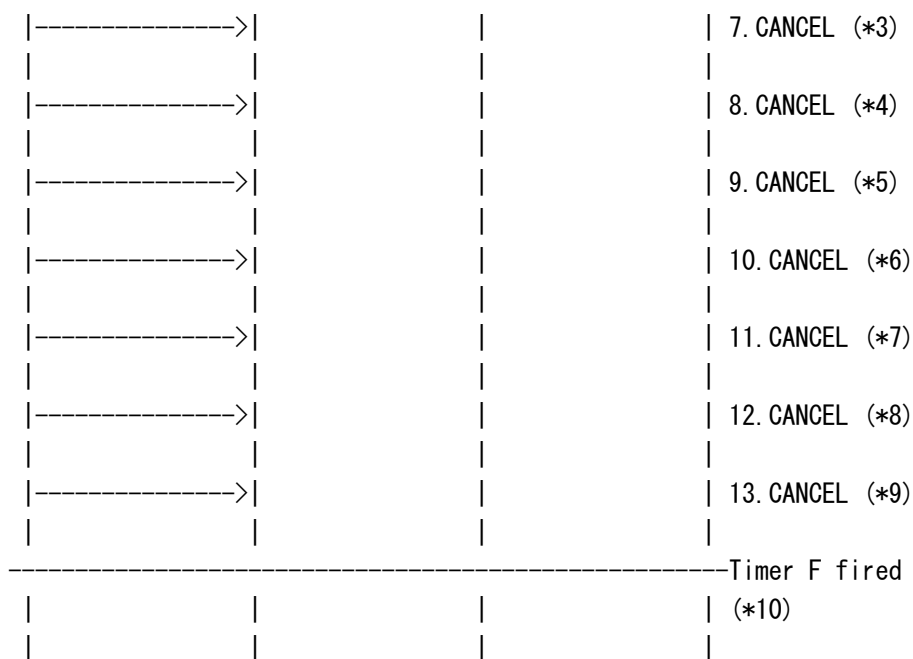
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Receive INVITE.
2. Send 100 Trying.
3. Send 180 Ringing.
4. Receive CANCEL.
5. Receive CANCEL. (\*1)
6. Receive CANCEL. (\*2)
7. Receive CANCEL. (\*3)
8. Receive CANCEL. (\*4)
9. Receive CANCEL. (\*5)
10. Receive CANCEL. (\*6)
11. Receive CANCEL. (\*7)
12. Receive CANCEL. (\*8)
13. Receive CANCEL. (\*9)
- (\*10)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1: CANCEL request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired.

[RFC3261-17-40][RFC3261-17-41][RFC3261-17-42]



\*2: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*3: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*4: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*5: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*6: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*7: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*8: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*9: CANCEL request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*10: before 200 response (Timer  $F(=64 \cdot T1)$  sec.) is already fired) from Proxy.

CANCEL request Must not be retransmitted. [RFC3261-17-44]

## [REFERENCE]

[RFC3261-17-40, 41, 42, 43, 44, 47, 48, 51, 52, 56, 57]

### 17.1.2.2 Formal Description

.....

The "Trying" state is entered when the TU initiates a new client transaction with a request. When entering this state, the client transaction SHOULD set timer F to fire in  $64 \cdot T1$  seconds. The request MUST be passed to the transport layer for transmission. If an unreliable transport is in use, the client transaction MUST set timer E to fire in  $T1$  seconds. If timer E fires while still in this state, the timer is reset, but this time with a value of  $\text{MIN}(2 \cdot T1, T2)$ . When the timer fires again, it is reset to a  $\text{MIN}(4 \cdot T1, T2)$ . This process continues so that retransmissions occur with an exponentially increasing interval that caps at  $T2$ . The default value of  $T2$  is 4s, and it represents the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately. For the default values of  $T1$  and  $T2$ , this results in intervals of 500 ms, 1 s, 2 s, 4 s, 4 s, 4 s, etc.

If Timer F fires while the client transaction is still in the "Trying" state, the client transaction SHOULD inform the TU about the timeout, and then it SHOULD enter the "Terminated" state. If a provisional response is received while in the "Trying" state, \*snip\* If a final response (status codes 200-699) is received while in the "Trying" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.

\*snip\* If timer F fires while in the "Proceeding" state, the TU MUST be informed of a timeout, and the client transaction MUST transition to the terminated state. If a final response (status codes 200-699) is received while in the "Proceeding" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.

\*snip\* The "Completed" state exists to buffer any additional response retransmissions that may be received (which is why the client transaction remains there only for unreliable transports).  $T4$  represents the amount of time the network will take to clear messages between client and server transactions.



The default value of T4 is 5s. A response is a retransmission when it matches the same transaction, using the rules specified in Section 17.1.3. If Timer K fires while in this state, the client transaction MUST transition to the "Terminated" state.

Once the transaction is in the terminated state, it MUST be destroyed immediately.

#### **4.3.5 UA-4-1-5 - Non-INVITE Client Transaction (Stop of retransmission of BYE upon Timer F fired)**

##### **[NAME]**

UA-4-1-5 - Non-INVITE Client Transaction (Stop of retransmission of BYE upon Timer F fired)

##### **[PURPOSE]**

Verify that a NUT properly processes non-INVITE client transaction of stopping retransmitting another BYE request when Timer F fires.

##### **[REQUIREMENT]**

NONE

##### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### **[PARAMETER]**

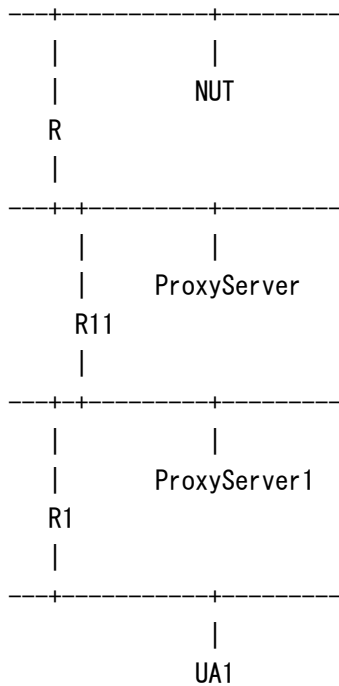
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

##### **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64



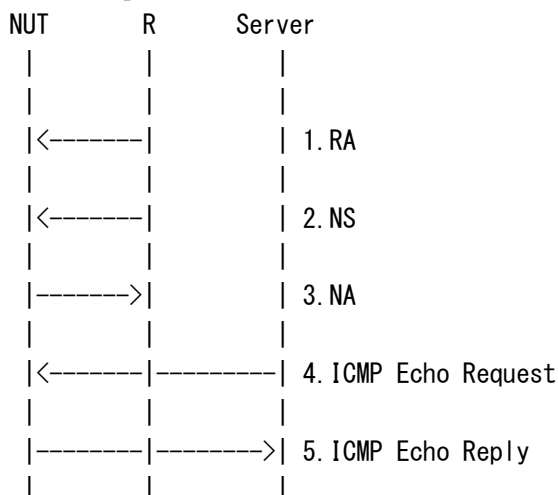
## [TOPOLOGY]



## [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

## [INITIALIZATION]



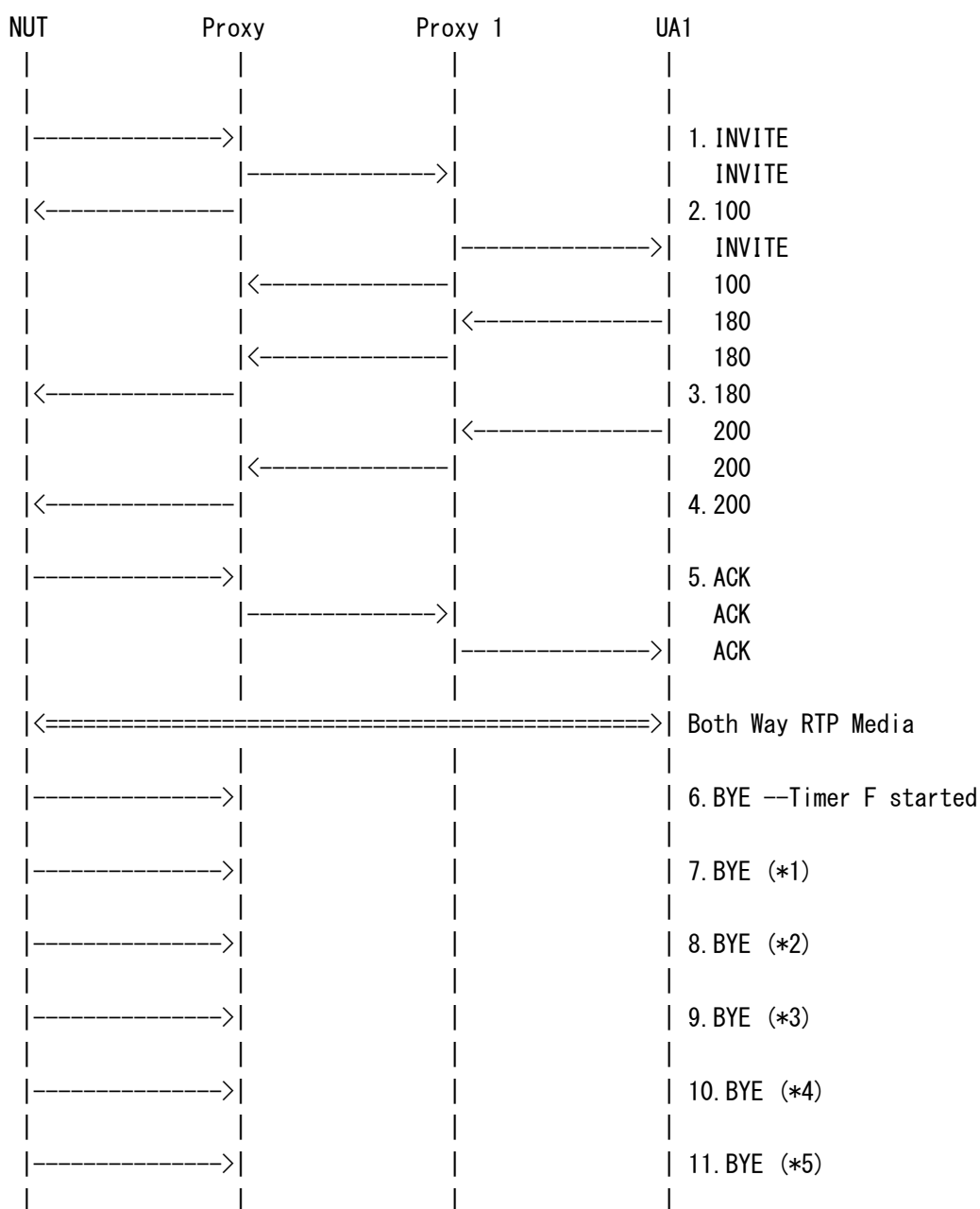
1. Send Router Advertisement.

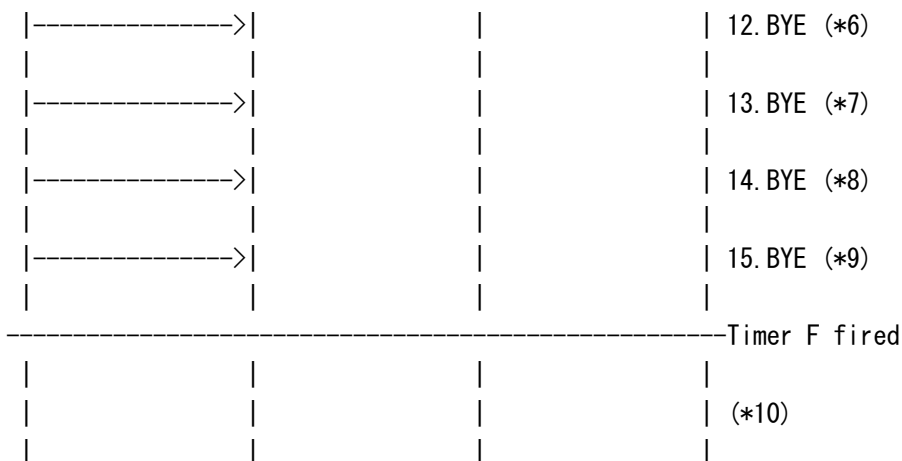
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Receive INVITE.
2. Send 100 Trying.
3. Send 180 Ringing.
4. Send 200 OK.
5. Receive ACK.
6. Receive BYE.
7. Receive BYE. (\*1)
8. Receive BYE. (\*2)
9. Receive BYE. (\*3)
10. Receive BYE. (\*4)
11. Receive BYE. (\*5)
12. Receive BYE. (\*6)
13. Receive BYE. (\*7)
14. Receive BYE. (\*8)
15. Receive BYE. (\*9)
- (\*10)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:BYE request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired.  
[RFC3261-17-40][RFC3261-17-41][RFC3261-17-42]

\*2:BYE request from NUT.

Must be retransmitted after MIN(2\*T1, T2) sec. [RFC3261-17-49, 50]



\*3:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*4:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*5:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*6:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*7:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*8:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*9:BYE request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*10:before 200 response(Timer F(=64\*T1 sec.) is already fired) from Proxy.

BYE request Must not be retransmitted. [RFC3261-17-44]

## [REFERENCE]

[RFC3261-17-40, 41, 42, 43, 44, 47, 48, 51, 52, 56, 57]

17.1.2.2 Formal Description

.....

The "Trying" state is entered when the TU initiates a new client transaction with a request. When entering this state, the client transaction SHOULD set timer F to fire in  $64 \cdot T1$  seconds. The request MUST be passed to the transport layer for transmission. If an unreliable transport is in use, the client transaction MUST set timer E to fire in  $T1$  seconds. If timer E fires while still in this state, the timer is reset, but this time with a value of  $\text{MIN}(2 \cdot T1, T2)$ . When the timer fires again, it is reset to a  $\text{MIN}(4 \cdot T1, T2)$ . This process continues so that retransmissions occur with an exponentially increasing interval that caps at  $T2$ . The default value of  $T2$  is 4s, and it represents the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately. For the default values of  $T1$  and  $T2$ , this results in intervals of 500 ms, 1 s, 2 s, 4 s, 4 s, 4 s, etc.

If Timer F fires while the client transaction is still in the "Trying" state, the client transaction SHOULD inform the TU about the timeout, and then it SHOULD enter the "Terminated" state. If a provisional response is received while in the "Trying" state, \*snip\* If a final response (status codes 200-699) is received while in the "Trying" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.

\*snip\* If timer F fires while in the "Proceeding" state, the TU MUST be informed of a timeout, and the client transaction MUST transition to the terminated state. If a final response (status codes 200-699) is received while in the "Proceeding" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.

\*snip\* The "Completed" state exists to buffer any additional response retransmissions that may be received (which is why the client transaction remains there only for unreliable transports).  $T4$  represents the amount of time the network will take to clear messages between client and server transactions. The default value of  $T4$  is 5s. A response is a retransmission when it matches the same transaction, using the rules specified in Section 17.1.3. If Timer K fires while in this state, the client transaction MUST transition to the "Terminated" state.



Once the transaction is in the terminated state, it **MUST** be destroyed immediately.

[RFC3261-10-10, 20]

## 10.2 Constructing the REGISTER Request

[RFC3261 Page 58 Paragraph 5]

UAs **MUST NOT** send a new registration (that is, containing new Contact header field values, as opposed to a retransmission) until they have received a final response from the registrar for the previous one or the previous REGISTER request has timed out.

### 10.2.7 Transmitting a Request

[RFC3261 Page 63 Paragraph 1]

If the transaction layer returns a timeout error because the REGISTER yielded no response, the UAC **SHOULD NOT** immediately re-attempt a registration to the same registrar.

An immediate re-attempt is likely to also timeout. Waiting some reasonable time interval for the conditions causing the timeout to be corrected reduces unnecessary load on the network. No specific interval is mandated.

### **4.3.6 UA-4-1-6 - Non-INVITE Client Transaction (Stop of retransmission of REGISTER upon Timer F fired)**

#### **[NAME]**

UA-4-1-6 - Non-INVITE Client Transaction (Stop of retransmission of REGISTER upon Timer F fired)

#### **[PURPOSE]**

Verify that a NUT properly processes non-INVITE client transaction of stopping retransmitting another REGISTER request when Timer F is fired.

#### **[REQUIREMENT]**

NONE

#### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### **[PARAMETER]**

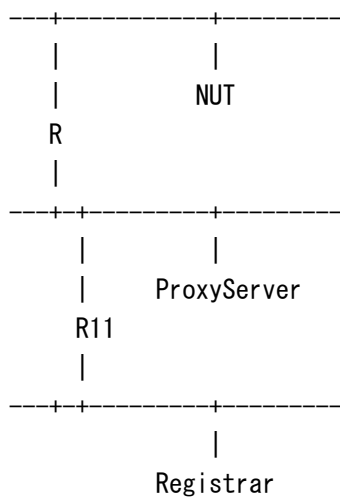


NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
T1	0.5sec
T2	4sec

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

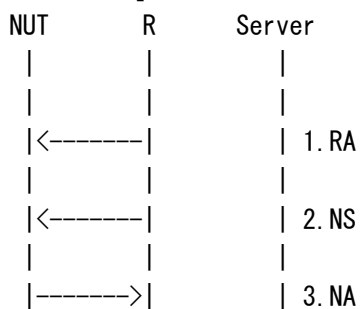
#### [TOPOLOGY]

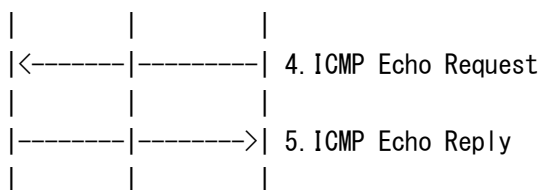


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip: reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

#### [INITIALIZATION]



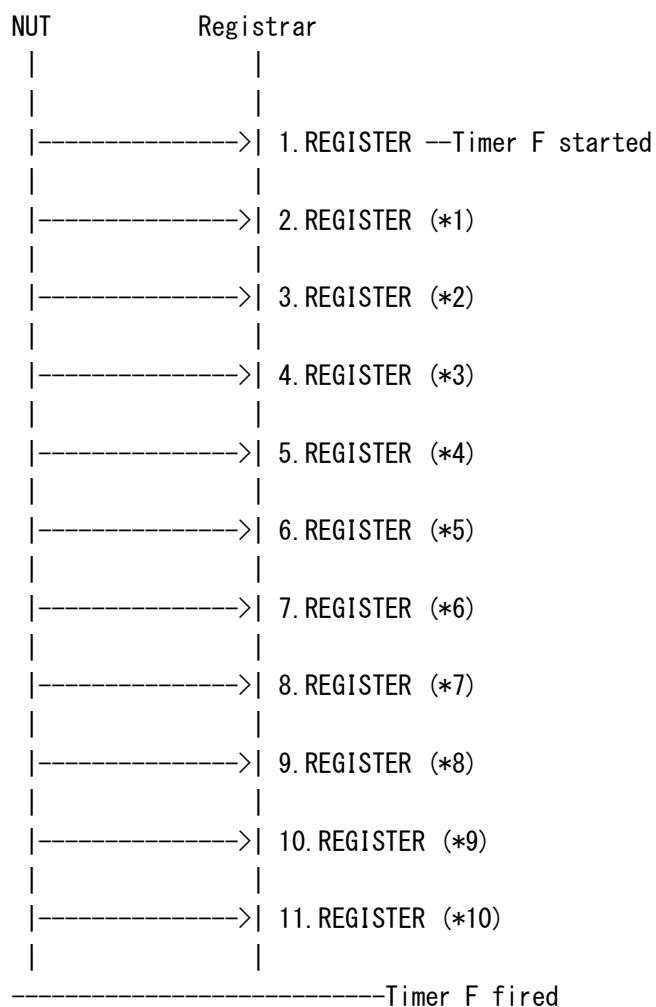


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





	(*11)

1. Receive REGISTER.
2. Receive REGISTER. (\*1)
3. Receive REGISTER. (\*2)
4. Receive REGISTER. (\*3)
5. Receive REGISTER. (\*4)
6. Receive REGISTER. (\*5)
7. Receive REGISTER. (\*6)
8. Receive REGISTER. (\*7)
9. Receive REGISTER. (\*8)
10. Receive REGISTER. (\*9)
11. Receive REGISTER. (\*10)
- (\*11)

### [OBSERVABLE RESULTS]

\*1:REGISTER request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired.  
[RFC3261-17-40][RFC3261-17-41][RFC3261-17-42]

\*2:REGISTER request from NUT.

Must be retransmitted after MIN(2\*T1, T2) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*3:REGISTER request from NUT.

Must be retransmitted after MIN(2\*T1, T2) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*4:REGISTER request from NUT.

Must be retransmitted after MIN(2\*T1, T2) sec. [RFC3261-17-49, 50]

\* Contact



Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*5:REGISTER request from NUT.

Must be retransmitted after MIN( $2 \cdot T1$ ,  $T2$ ) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*6:REGISTER request from NUT.

Must be retransmitted after MIN( $2 \cdot T1$ ,  $T2$ ) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*7:REGISTER request from NUT.

Must be retransmitted after MIN( $2 \cdot T1$ ,  $T2$ ) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*8:REGISTER request from NUT.

Must be retransmitted after MIN( $2 \cdot T1$ ,  $T2$ ) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*9:REGISTER request from NUT.

Must be retransmitted after MIN( $2 \cdot T1$ ,  $T2$ ) sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]



\*10:REGISTER request from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\* Contact

Must be the same value as all of REGISTER request, If Contact header field exists. [RFC3261-10-10]

\*11:after Timer F( $=64 \cdot T1$  sec.) fired.

Another REGISTER request should not be transmitted.(No specific interval is mandated.) [RFC3261-10-20, RFC3261-17-44]

## [REFERENCE]

[RFC3261-17-40, 41, 42, 43, 44, 47, 48, 51, 52, 56, 57]

### 17.1.2.2 Formal Description

.....

The "Trying" state is entered when the TU initiates a new client transaction with a request. When entering this state, the client transaction SHOULD set timer F to fire in  $64 \cdot T1$  seconds. The request MUST be passed to the transport layer for transmission. If an unreliable transport is in use, the client transaction MUST set timer E to fire in  $T1$  seconds. If timer E fires while still in this state, the timer is reset, but this time with a value of  $\text{MIN}(2 \cdot T1, T2)$ . When the timer fires again, it is reset to a  $\text{MIN}(4 \cdot T1, T2)$ . This process continues so that retransmissions occur with an exponentially increasing interval that caps at  $T2$ . The default value of  $T2$  is 4s, and it represents the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately. For the default values of  $T1$  and  $T2$ , this results in intervals of 500 ms, 1 s, 2 s, 4 s, 4 s, 4 s, etc.

If Timer F fires while the client transaction is still in the "Trying" state, the client transaction SHOULD inform the TU about the timeout, and then it SHOULD enter the "Terminated" state. If a provisional response is received while in the "Trying" state,  
\*snip\* If a final response (status codes 200-699) is received while in the "Trying" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.



\*snip\* If timer F fires while in the "Proceeding" state, the TU MUST be informed of a timeout, and the client transaction MUST transition to the terminated state. If a final response (status codes 200-699) is received while in the "Proceeding" state, the response MUST be passed to the TU, and the client transaction MUST transition to the "Completed" state.

\*snip\* The "Completed" state exists to buffer any additional response retransmissions that may be received (which is why the client transaction remains there only for unreliable transports). T4 represents the amount of time the network will take to clear messages between client and server transactions. The default value of T4 is 5s. A response is a retransmission when it matches the same transaction, using the rules specified in Section 17.1.3. If Timer K fires while in this state, the client transaction MUST transition to the "Terminated" state.

Once the transaction is in the terminated state, it MUST be destroyed immediately.

[RFC3261-10-10, 20]

#### 10.2 Constructing the REGISTER Request

[RFC3261 Page 58 Paragraph 5]

UAs MUST NOT send a new registration (that is, containing new Contact header field values, as opposed to a retransmission) until they have received a final response from the registrar for the previous one or the previous REGISTER request has timed out.

##### 10.2.7 Transmitting a Request

[RFC3261 Page 63 Paragraph 1]

If the transaction layer returns a timeout error because the REGISTER yielded no response, the UAC SHOULD NOT immediately re-attempt a registration to the same registrar.

An immediate re-attempt is likely to also timeout. Waiting some reasonable time interval for the conditions causing the timeout to be corrected reduces unnecessary load on the network. No specific interval is mandated.

### 4.3.7 UA-4-1-7 - Non-INVITE Client Transaction (Receipt of 100 response)



## to CANCEL and reset of Timer E with T2 )

### [NAME]

UA-4-1-7 - Non-INVITE Client Transaction (Receipt of 100 response to CANCEL and reset of Timer E with T2 )

### [PURPOSE]

Verify that a NUT resets Timer E with T2 when receiving a 100 response to previous CANCEL request.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

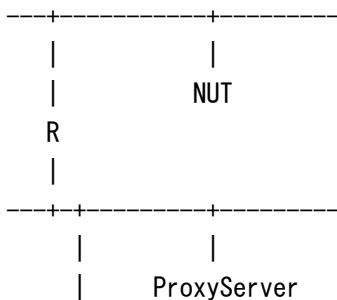
### [PARAMETER]

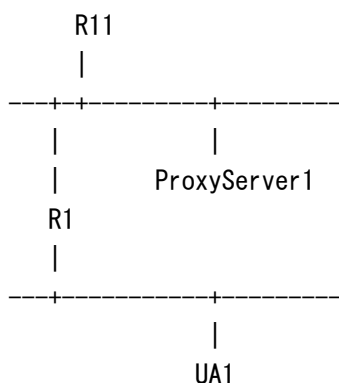
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]

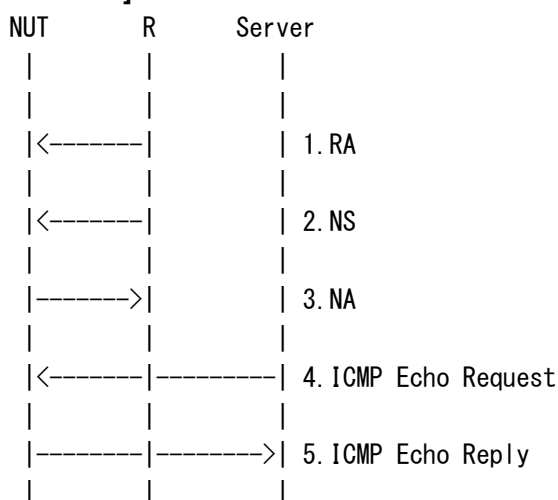




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.



1. Receive INVITE.
2. Send 180 Ringing.
3. Receive CANCEL.
4. Receive CANCEL. (\*1)
5. Receive CANCEL. (\*2)
6. Send 100 Trying.(for CANCEL)
7. Receive CANCEL.  
(\*3)
8. Receive CANCEL.
9. Send 200 OK.
10. Send 487 Request Terminated.
11. Receive ACK.

#### **[OBSERVABLE RESULTS]**

\*\* this scenario checks only timing (message format is not checked)

\*1:CANCEL request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired. [RFC3261-17-42]

\*2:CANCEL request from NUT.

Must be retransmitted in the interval of MIN(2\*T1,T2) sec. [RFC3261-17-49, 50]

\*3:after 100 response from Proxy.

Should be retransmitted after T2 sec. [RFC3261-17-49][RFC3261-17-50]

#### **[REFERENCE]**

[RFC3261-17-45, 46, 49, 50]

17.1.2.2 Formal Description

.....

\*snip\* If a

provisional response is received while in the "Trying" state, the response MUST be passed to the TU, and then the client transaction SHOULD move to the "Proceeding" state. \*snip\*



.....

If Timer E fires while in the "Proceeding" state, the request **MUST** be passed to the transport layer for retransmission, and Timer E **MUST** be reset with a value of T2 seconds.

#### 4.3.8 UA-4-1-8 - Non-INVITE Client Transaction (Receipt of 100 response to BYE and reset of Timer E with T2)

##### [NAME]

UA-4-1-8 - Non-INVITE Client Transaction (Receipt of 100 response to BYE and reset of Timer E with T2)

##### [PURPOSE]

Verify that a NUT resets Timer E with T2 when receiving a 100 response to previous BYE request.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

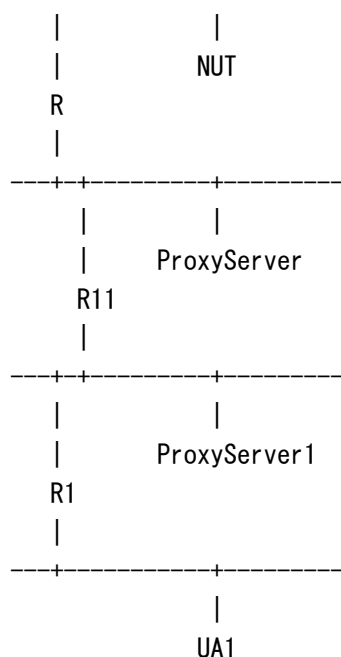
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

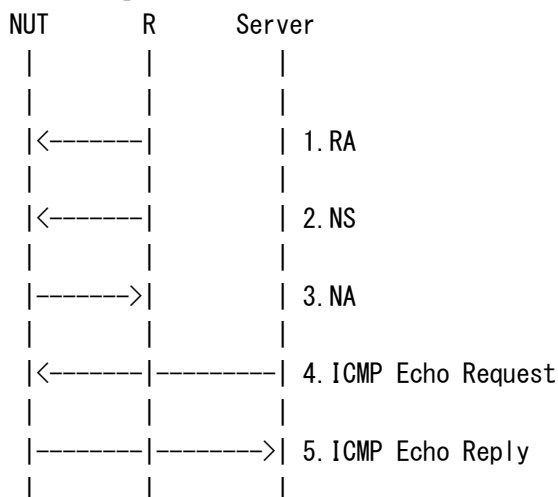
-----+-----+-----



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



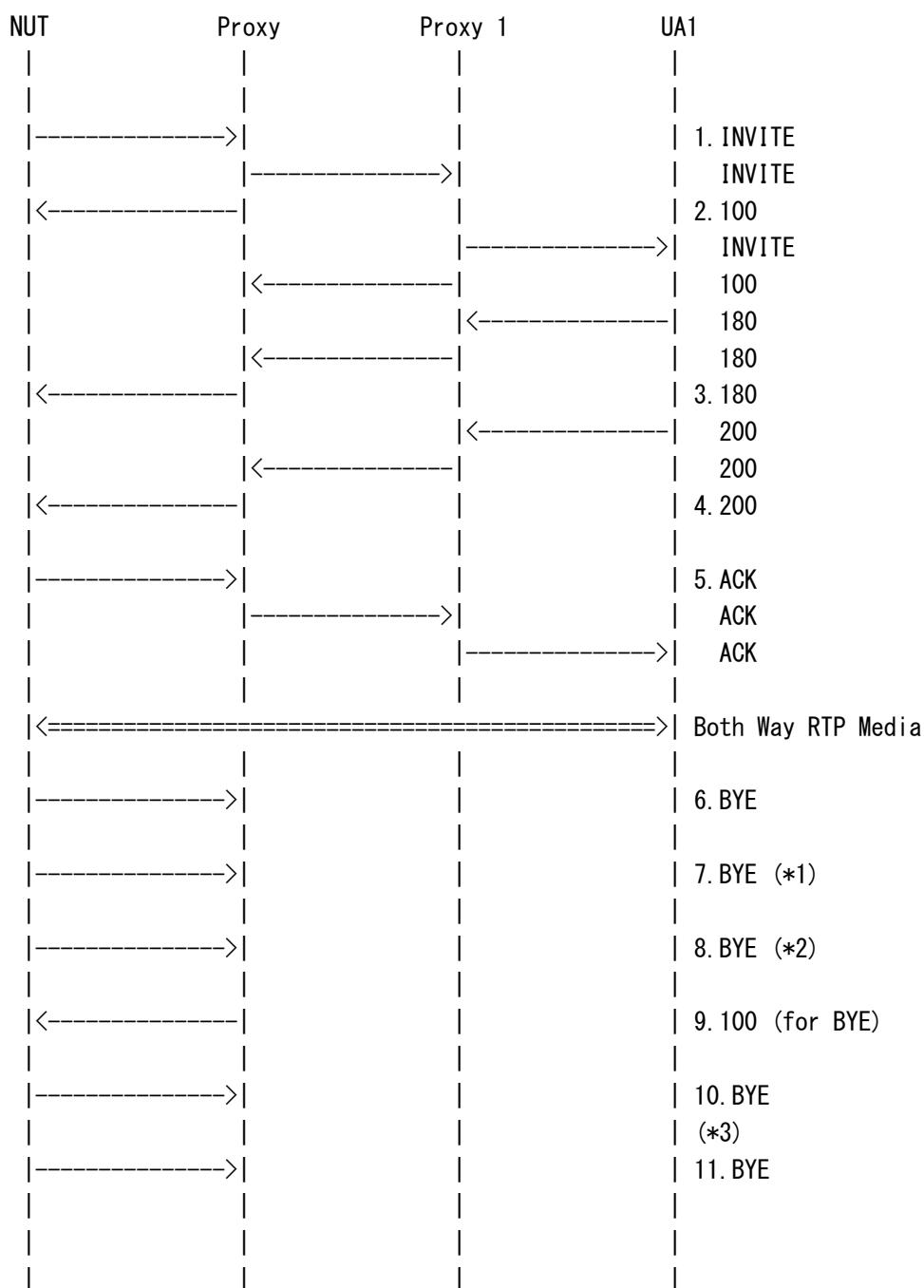
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.

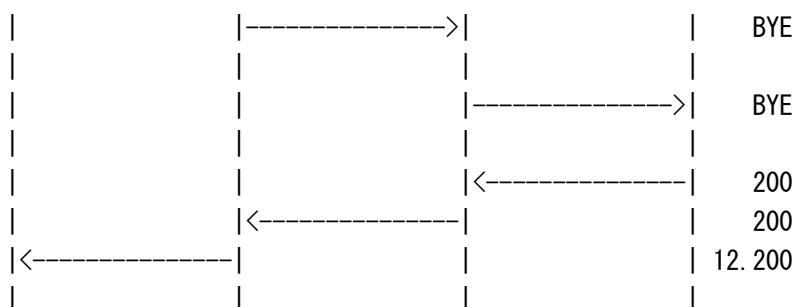
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Receive INVITE.
2. Send 100 Trying.
3. Send 180 Ringing.
4. Send 200 OK.
5. Receive ACK.
6. Receive BYE.
7. Receive BYE. (\*1)
8. Receive BYE. (\*2)
9. Send 100 Trying (for BYE).
10. Receive BYE.  
(\*3)
11. Receive BYE.
12. Send 200 OK.

### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:BYE request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired. [RFC3261-17-42]

\*2:BYE request from NUT.

Must be retransmitted in the interval of MIN(2\*T1,T2) sec. [RFC3261-17-49, 50]

\*3:after 100 response from Proxy.

Should be retransmitted after T2 sec. [RFC3261-17-49][RFC3261-17-50]

### [REFERENCE]

[RFC3261-17-45, 46, 49, 50]

### 17.1.2.2 Formal Description

.....

\*snip\* If a provisional response is received while in the "Trying" state, the response MUST be passed to the TU, and then the client transaction SHOULD move to the "Proceeding" state. \*snip\*

.....

If Timer E fires while in the "Proceeding" state, the request MUST be passed to the transport layer for retransmission, and Timer E MUST be reset with a value of T2 seconds.

### 4.3.9 UA-4-1-9 - Non-INVITE Client Transaction (Receipt of 100 response to REGISTER and reset of Timer E with T2)

#### [NAME]

UA-4-1-9 - Non-INVITE Client Transaction (Receipt of 100 response to REGISTER and reset of Timer E with T2)

#### [PURPOSE]

Verify that a NUT resets Timer E with T2 when receiving a 100 response to previous REGISTER request.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

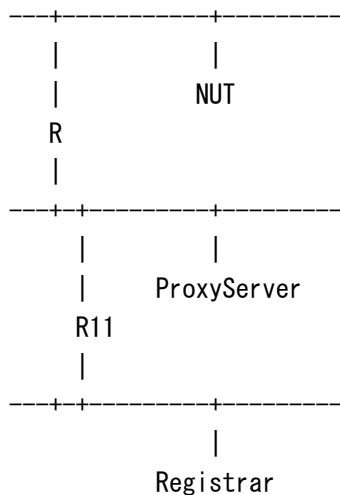
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
T1	0.5sec
T2	4sec

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64

ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

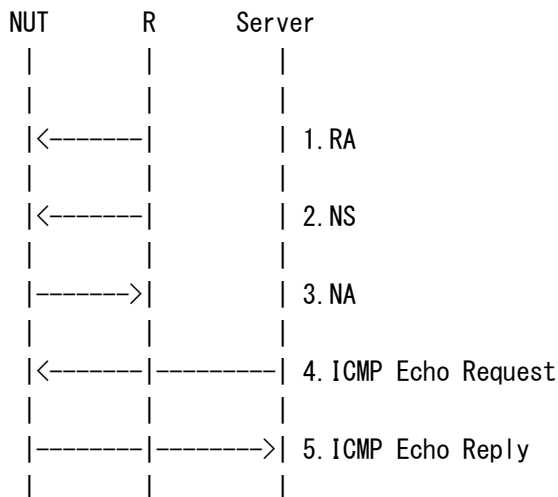
### [TOPOLOGY]



### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip: reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

### [INITIALIZATION]



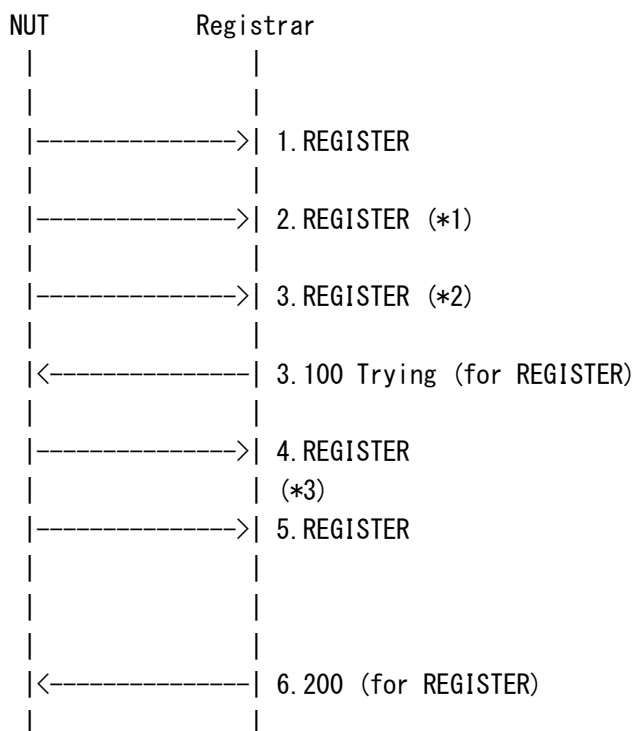
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.

## 5. Receive ICMP Echo Reply.

### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.



1. Receive REGISTER.
2. Receive REGISTER. (\*1)
3. Receive REGISTER. (\*2)
4. Send 100 Trying (for REGISTER). (\*3)
5. Receive REGISTER.
6. Receive REGISTER.
7. Send 200 OK (for REGISTER).

### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:REGISTER request from NUT.

Must be retransmitted after Timer E(= T1 sec.) fired. [RFC3261-17-42]



\*2:REGISTER request from NUT.

Must be retransmitted in the interval of  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261-17-49, 50]

\*3:after 100 response from Proxy.

Should be retransmitted after  $T2$  sec. [RFC3261-17-49][RFC3261-17-50]

#### [REFERENCE]

[RFC3261-17-45, 46, 49, 50]

#### 17.1.2.2 Formal Description

.....

\*snip\* If a provisional response is received while in the "Trying" state, the response **MUST** be passed to the TU, and then the client transaction **SHOULD** move to the "Proceeding" state. \*snip\*

.....

If Timer E fires while in the "Proceeding" state, the request **MUST** be passed to the transport layer for retransmission, and Timer E **MUST** be reset with a value of  $T2$  seconds.

### 4.3.10 UA-4-1-10 - INVITE Server Transaction (Stop retransmission after Timer H fired)

#### [NAME]

UA-4-1-10 - INVITE Server Transaction (Stop retransmission after Timer H fired)

#### [PURPOSE]

Verify that a NUT properly stops retransmitting another 4xx-6xx response after Timer H is fired.

#### [REQUIREMENT]

NONE

#### [TARGET]



SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

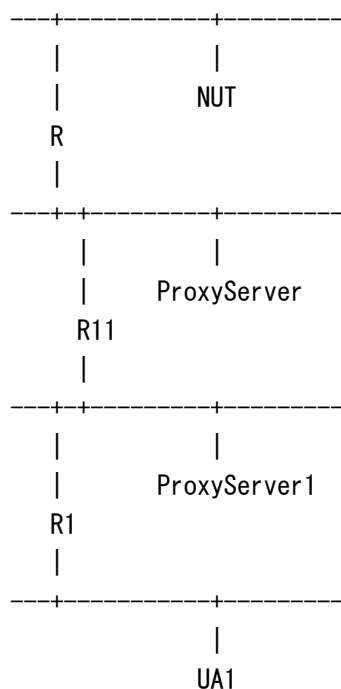
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

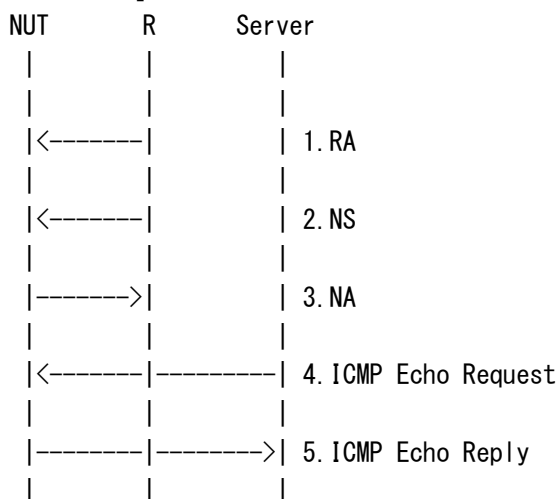


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

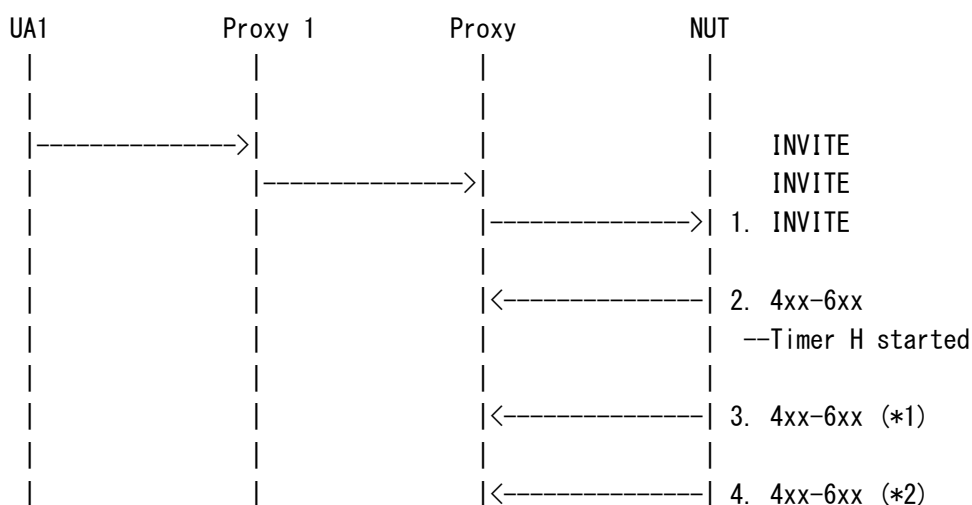


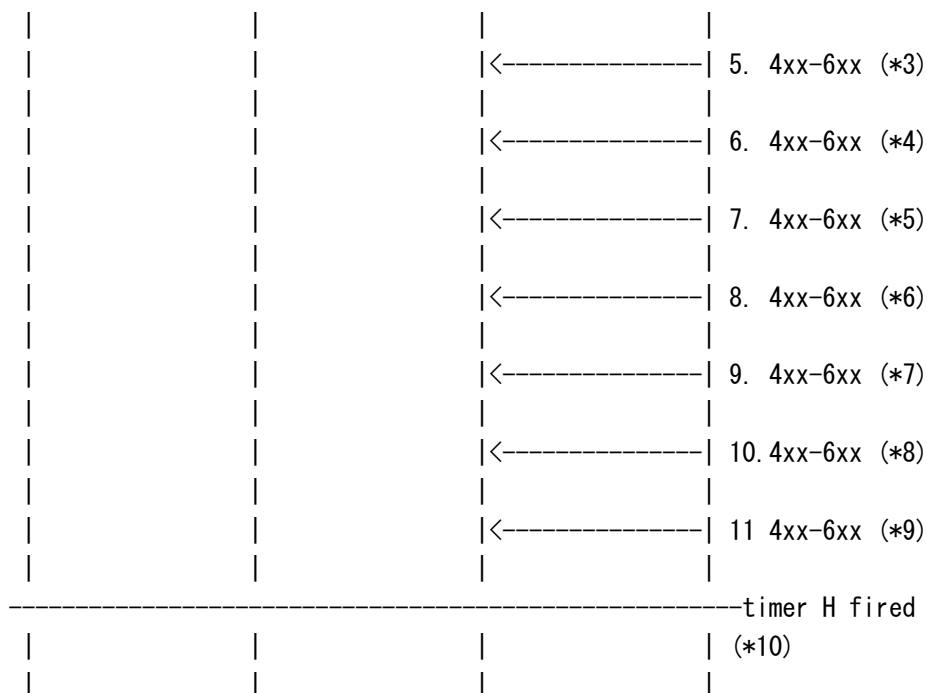
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Send INVITE.
2. Receive 4xx-6xx.
3. Receive 4xx-6xx. (\*1)
4. Receive 4xx-6xx. (\*2)
5. Receive 4xx-6xx. (\*3)
6. Receive 4xx-6xx. (\*4)
7. Receive 4xx-6xx. (\*5)
8. Receive 4xx-6xx. (\*6)
9. Receive 4xx-6xx. (\*7)
10. Receive 4xx-6xx. (\*8)
11. Receive 4xx-6xx. (\*9)
- (\*10)

#### [OBSERVABLE RESULTS]

- \*\* this scenario checks only timing (message format is not checked)
- \*\* Target receives INVITE with To-tag, and Tester expects 481 response.
- \*\* But target may do any way that like for sending 4xx-6xx response.

\*1:4xx-6xx response from NUT.

Must be retransmitted after Timer G(= T1 sec.) fired. [RFC3261-17-67, RFC3261-17-68]



\*2: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*3: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*4: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*5: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*6: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*7: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*8: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*9: 4xx-6xx response from NUT.

Must be retransmitted after  $\text{MIN}(2 \cdot T1, T2)$  sec. [RFC3261 17.2.1]

\*10: after Timer H( $=64 \cdot T1$  sec.) fired.

4xx-6xx response Must not be retransmitted. [RFC3261-17-69, RFC3261-17-72, RFC3261-17-73]

## [REFERENCE]



[RFC3261-17-67, 68, 69, 70, 72, 73]

#### 17.2.1 INVITE Server Transaction

.....

While in the "Proceeding" state, if the TU passes a response with status code from 300 to 699 to the server transaction, the response **MUST** be passed to the transport layer for transmission, and the state machine **MUST** enter the "Completed" state. For unreliable transports, timer G is set to fire in T1 seconds, and is not set to fire for reliable transports.

This is a change from RFC 2543, where responses were always retransmitted, even over reliable transports.

When the "Completed" state is entered, timer H **MUST** be set to fire in  $64 \cdot T1$  seconds for all transports. Timer H determines when the server transaction abandons retransmitting the response. Its value is chosen to equal Timer B, the amount of time a client transaction will continue to retry sending a request. If timer G fires, the response is passed to the transport layer once more for retransmission, and timer G is set to fire in  $\text{MIN}(2 \cdot T1, T2)$  seconds. From then on, when timer G fires, the response is passed to the transport again for transmission, and timer G is reset with a value that doubles, unless that value exceeds T2, in which case it is reset with the value of T2. This is identical to the retransmit behavior for requests in the "Trying" state of the non-INVITE client transaction. Furthermore, while in the "Completed" state, if a request retransmission is received, the server **SHOULD** pass the response to the transport for retransmission.

.....

If timer H fires while in the "Completed" state, it implies that the ACK was never received. In this case, the server transaction **MUST** transition to the "Terminated" state, and **MUST** indicate to the TU that a transaction failure has occurred.

#### 4.3.11 UA-4-1-11 - INVITE Server Transaction (Response after Timer H fired)

[NAME]

UA-4-1-11 - INVITE Server Transaction (Response after Timer H fired)

## [PURPOSE]

Verify that a NUT doesn't retransmit 4xx-6xx response to the requests that has the same to-tag as one of response after Timer H is fired. Also, verify that the UA retransmits 4xx-6xx response to the requests, when the to-tag of that request is different from that before time-out of Timer H, after Timer H is fired..

## [REQUIREMENT]

NONE

## [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

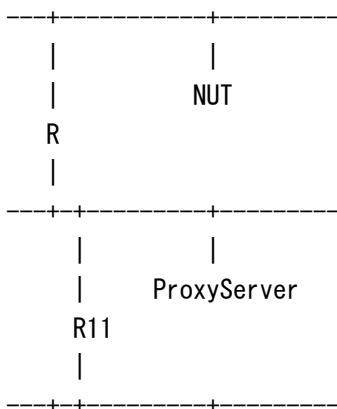
## [PARAMETER]

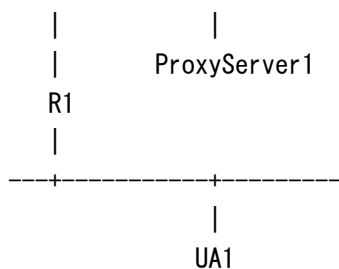
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

## [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

## [TOPOLOGY]

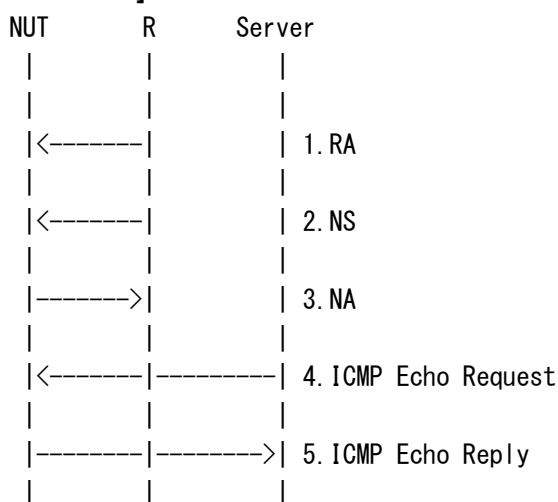




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

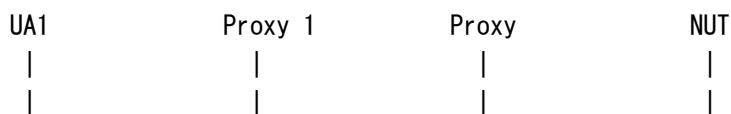


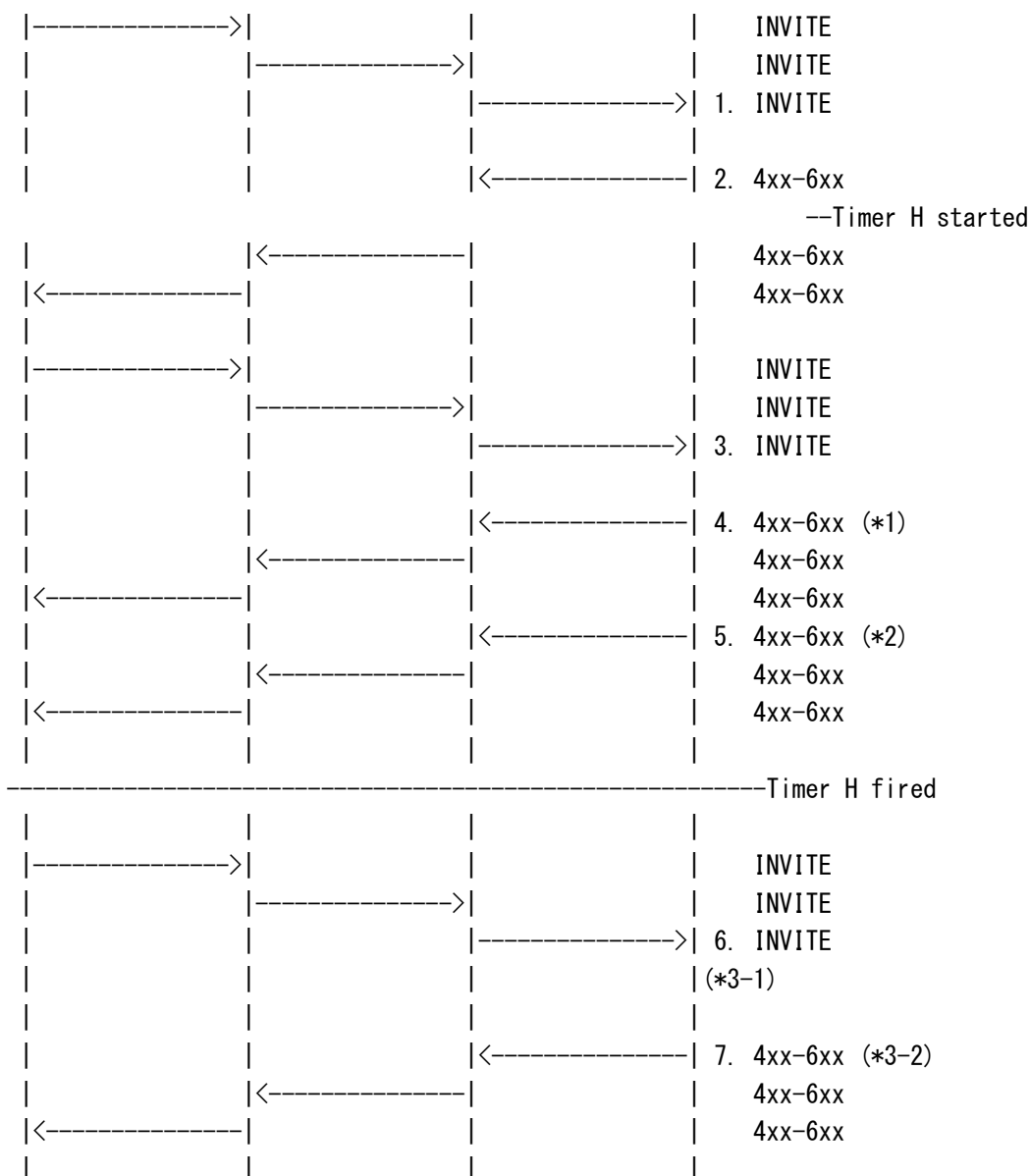
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Send INVITE.
2. Receive 4xx-6xx.
3. Send INVITE.
4. Receive 4xx-6xx. (\*1)
5. Receive 4xx-6xx. (\*2)
6. Send INVITE.  
(\*3-1)
7. Receive 4xx-6xx. (\*3-2)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked, except 1xx





response)

\*1:4xx-6xx response from NUT.

Must be retransmitted after Timer G(= T1 sec.) fired. [RFC3261-17-67, RFC3261-17-68]

\*2:4xx-6xx response from NUT.

Must be retransmitted in the interval of MIN(2\*T1,T2) sec. [RFC3261 17.2.1]

\*3-1:after Timer H(=64\*T1 sec.) fired.

4xx-6xx response Must not be retransmitted. [RFC3261-17-69, RFC3261-17-72, RFC3261-17-73]

\*3-2:4xx-6xx response from NUT.

To tag Must be different from retransmitted 4xx-6xx response before Timer H fires. [RFC3261-17-72, 73]

## **[REFERENCE]**

[RFC3261-17-67, 68, 69, 70, 72, 73]

17.2.1 INVITE Server Transaction

.....

While in the "Proceeding" state, if the TU passes a response with status code from 300 to 699 to the server transaction, the response MUST be passed to the transport layer for transmission, and the state machine MUST enter the "Completed" state. For unreliable transports, timer G is set to fire in T1 seconds, and is not set to fire for reliable transports.

This is a change from RFC 2543, where responses were always retransmitted, even over reliable transports.

When the "Completed" state is entered, timer H MUST be set to fire in 64\*T1 seconds for all transports. Timer H determines when the server transaction abandons retransmitting the response. Its value is

chosen to equal Timer B, the amount of time a client transaction will continue to retry sending a request. If timer G fires, the response is passed to the transport layer once more for retransmission, and timer G is set to fire in  $\text{MIN}(2 \cdot T1, T2)$  seconds. From then on, when timer G fires, the response is passed to the transport again for transmission, and timer G is reset with a value that doubles, unless that value exceeds T2, in which case it is reset with the value of T2. This is identical to the retransmit behavior for requests in the "Trying" state of the non-INVITE client transaction. Furthermore, while in the "Completed" state, if a request retransmission is received, the server SHOULD pass the response to the transport for retransmission.

.....

If timer H fires while in the "Completed" state, it implies that the ACK was never received. In this case, the server transaction MUST transition to the "Terminated" state, and MUST indicate to the TU that a transaction failure has occurred.

#### 4.3.12 UA-4-1-12 - INVITE Server Transaction (Stop of retransmission of 4xx-6xx response upon receipt of ACK)

##### [NAME]

UA-4-1-12 - INVITE Server Transaction (Stop of retransmission of 4xx-6xx response upon receipt of ACK)

##### [PURPOSE]

Verify that a NUT stops retransmitting another 4xx-6xx response when receiving ACK.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

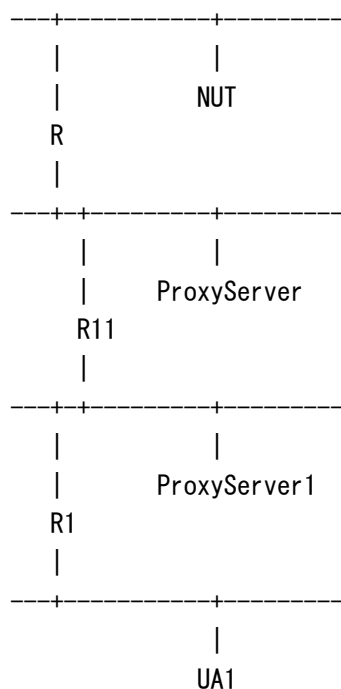
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

T1	0.5sec
T2	4sec

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

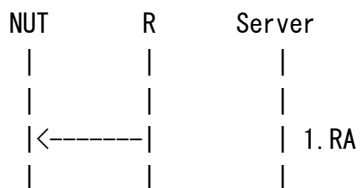
#### [TOPOLOGY]

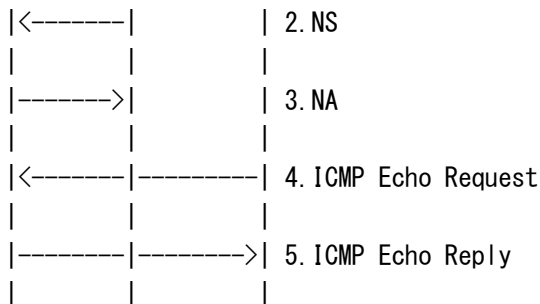


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



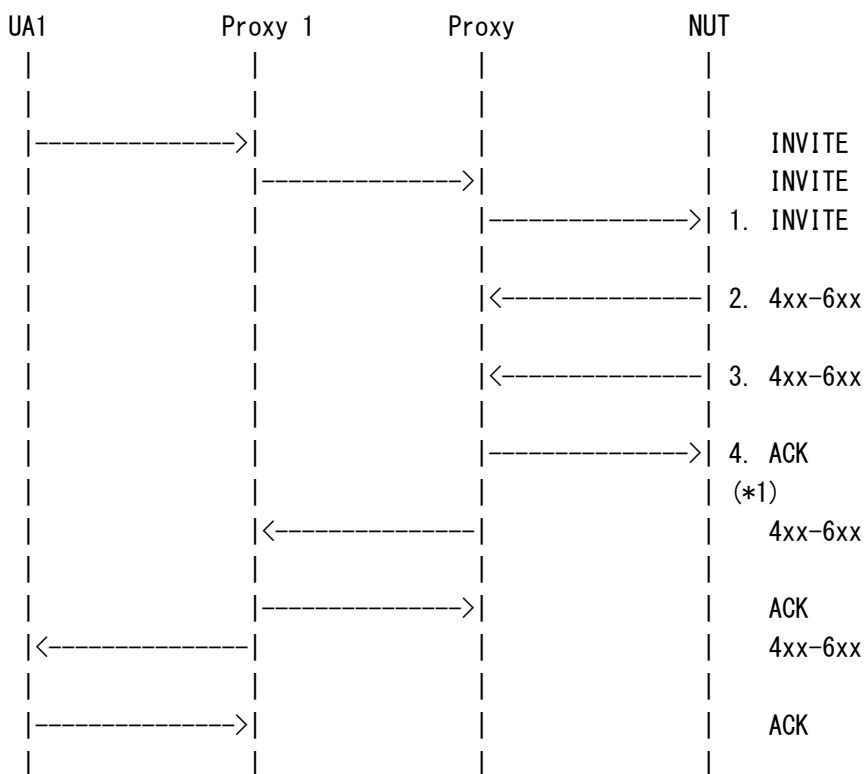


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.



1. Send INVITE.
2. Receive 4xx-6xx.



3. Receive 4xx-6xx.

4. Send ACK.

(\*1)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*\* Target receives INVITE with To-tag, and Tester expects 481 response.

\*\* But target may do any way that like for sending 4xx-6xx response.

\*1:after ACK request from Proxy.

4xx-6xx response Must not be retransmitted. [RFC3261-17-71]

#### [REFERENCE]

[RFC3261-17-67, 68, 71]

17.2.1 INVITE Server Transaction

.....

While in the "Proceeding" state, if the TU passes a response with status code from 300 to 699 to the server transaction, the response MUST be passed to the transport layer for transmission, and the state machine MUST enter the "Completed" state. For unreliable transports, timer G is set to fire in T1 seconds, and is not set to fire for reliable transports.

.....

If an ACK is received while the server transaction is in the "Completed" state, the server transaction MUST transition to the "Confirmed" state. As Timer G is ignored in this state, any retransmissions of the response will cease.

#### **4.3.13 UA-4-1-13 - Non-INVITE Server Transaction (Stop of retransmission of CANCEL after Timer J fired)**

##### [NAME]

UA-4-1-13 - Non-INVITE Server Transaction (stop of retransmission of CANCEL after Timer J fired)

##### [PURPOSE]

Verify that a NUT properly stops retransmitting another CANCEL request after Timer J is

fired.

# **[REQUIREMENT]**

NONE

# **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

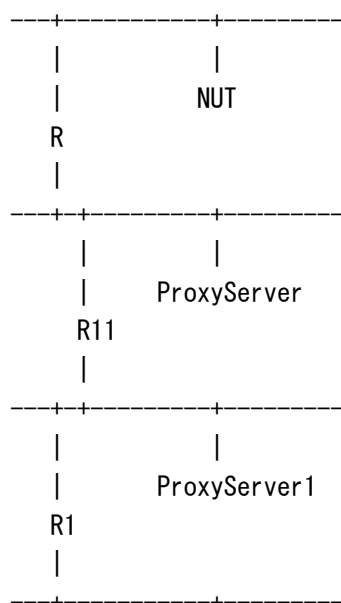
# **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

# **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# **[TOPOLOGY]**

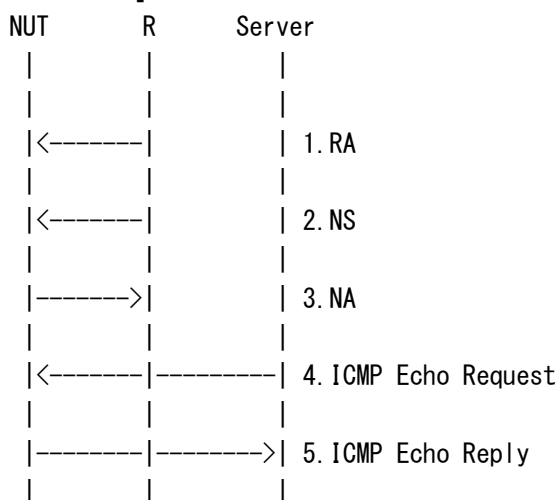


UA1

### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

### [INITIALIZATION]

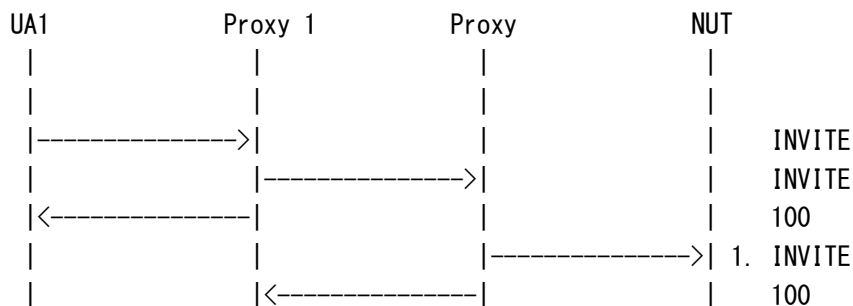


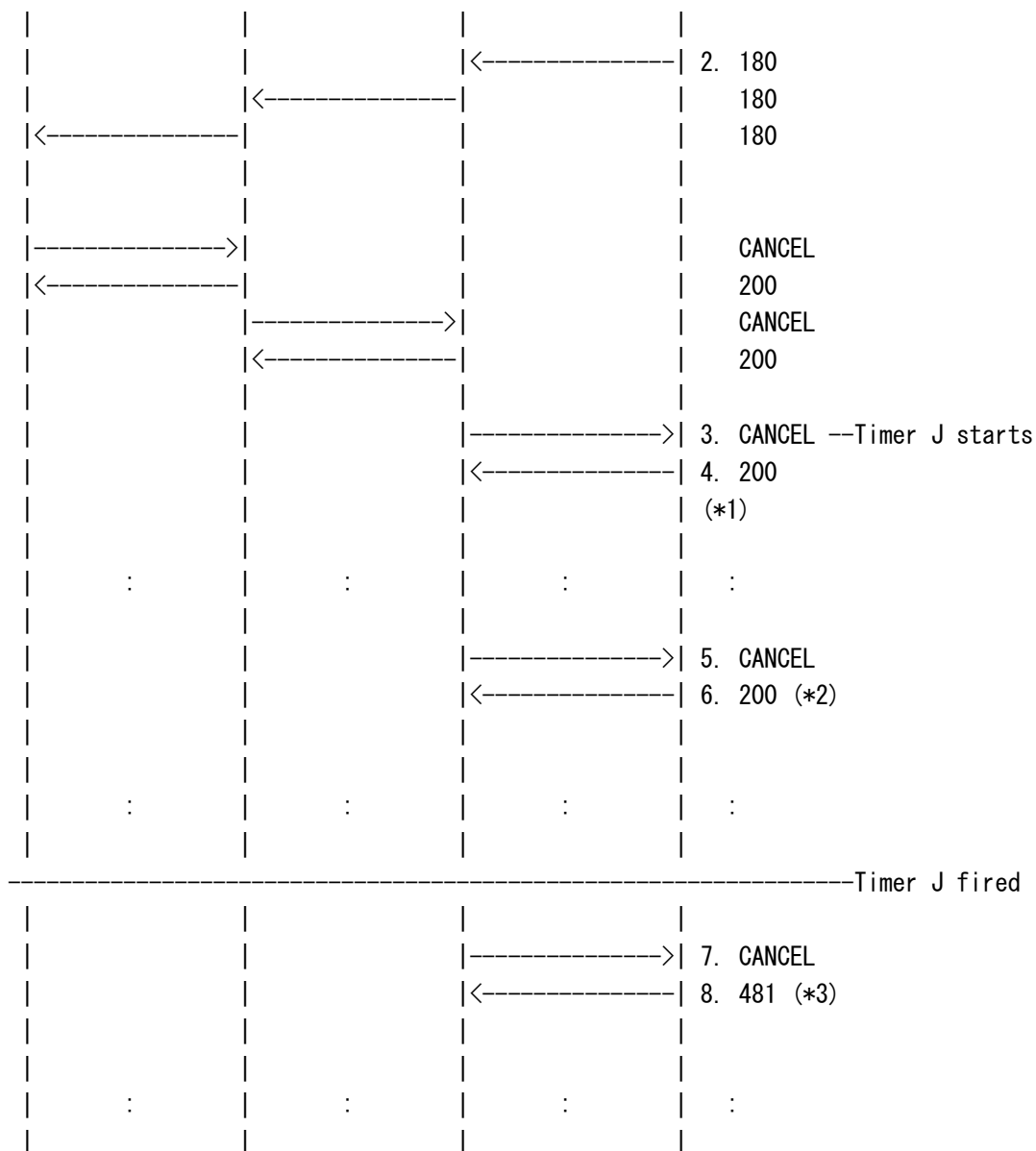
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.





1. Send INVITE.
  2. Receive 180 Ringing.
  3. Send CANCEL. -- Before Timer J \* Timer J has started counting down...
  4. Receive 200 OK.
  5. Send CANCEL. -- every 2 seconds (Note: Timer J doesn't reset)
  6. Receive 200 OK. (\*1)
  7. Send CANCEL -- After Timer J terminates.
  8. Receive 481 Call/Transaction Does Not Exist. (\*2)
- : repeat 7-8 every 2 seconds (Note: Timer J doesn't reset)
- finish J + 30sec





### [OBSERVABLE RESULTS]

- \*\* this scenario checks only timing (message format is not checked, except lxx response)
- \*\* response(481 is used below) may be 300-699 response.

\*1:after 200 response from NUT.

Must not retransmit further 200 response. [RFC3261-17-83, 84, 85, 86]

\*2:200 response from NUT. (CANCEL is sent every 2 seconds)

Must send 200 response. [RFC3261-17-80][RFC3261-17-81][RFC3261-17-82]

Must send response before Timer J(=64\*T1 sec.) fires. [RFC3261-17-83]

\*3:after Timer J fires.(CANCEL is sent every 2 seconds)

Must send 481 response. [RFC3261-17-85][RFC3261-17-86]

### [REFERENCE]

[RFC3261-17-83, 84, 85, 86]

#### 17.2.2 Non-INVITE Server Transaction

When the server transaction enters the "Completed" state, it MUST set Timer J to fire in 64\*T1 seconds for unreliable transports, and zero seconds for reliable transports. While in the "Completed" state, the server transaction MUST pass the final response to the transport layer for retransmission whenever a retransmission of the request is received. Any other final responses passed by the TU to the server transaction MUST be discarded while in the "Completed" state. The server transaction remains in this state until Timer J fires, at which point it MUST transition to the "Terminated" state.

The server transaction MUST be destroyed the instant it enters the "Terminated" state.

#### **4.3.14 UA-4-1-14 - Non-INVITE Server Transaction (Stop of retransmission of BYE)**

##### [NAME]

UA-4-1-14 - Non-INVITE Server Transaction (Stop of retransmission of BYE)

**[PURPOSE]**

Verify that a NUT properly stops retransmitting another BYE request after Timer J is fired.

**[REQUIREMENT]**

NONE

**[TARGET]**

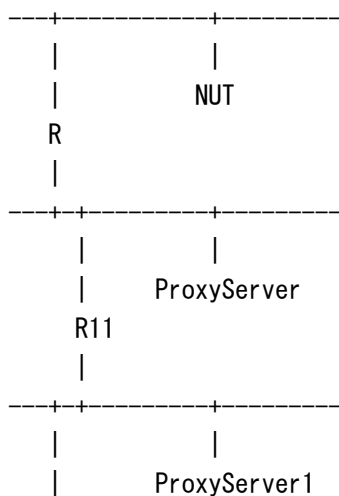
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

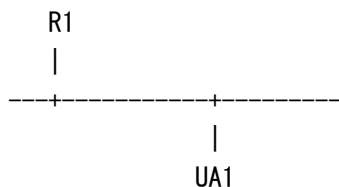
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec
T2	4sec

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

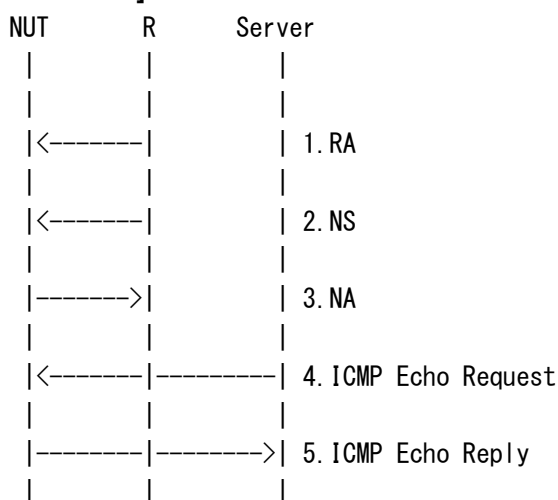
**[TOPOLOGY]**



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

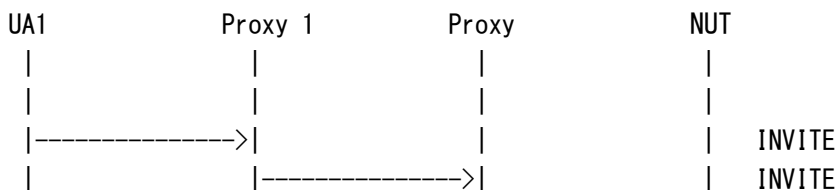


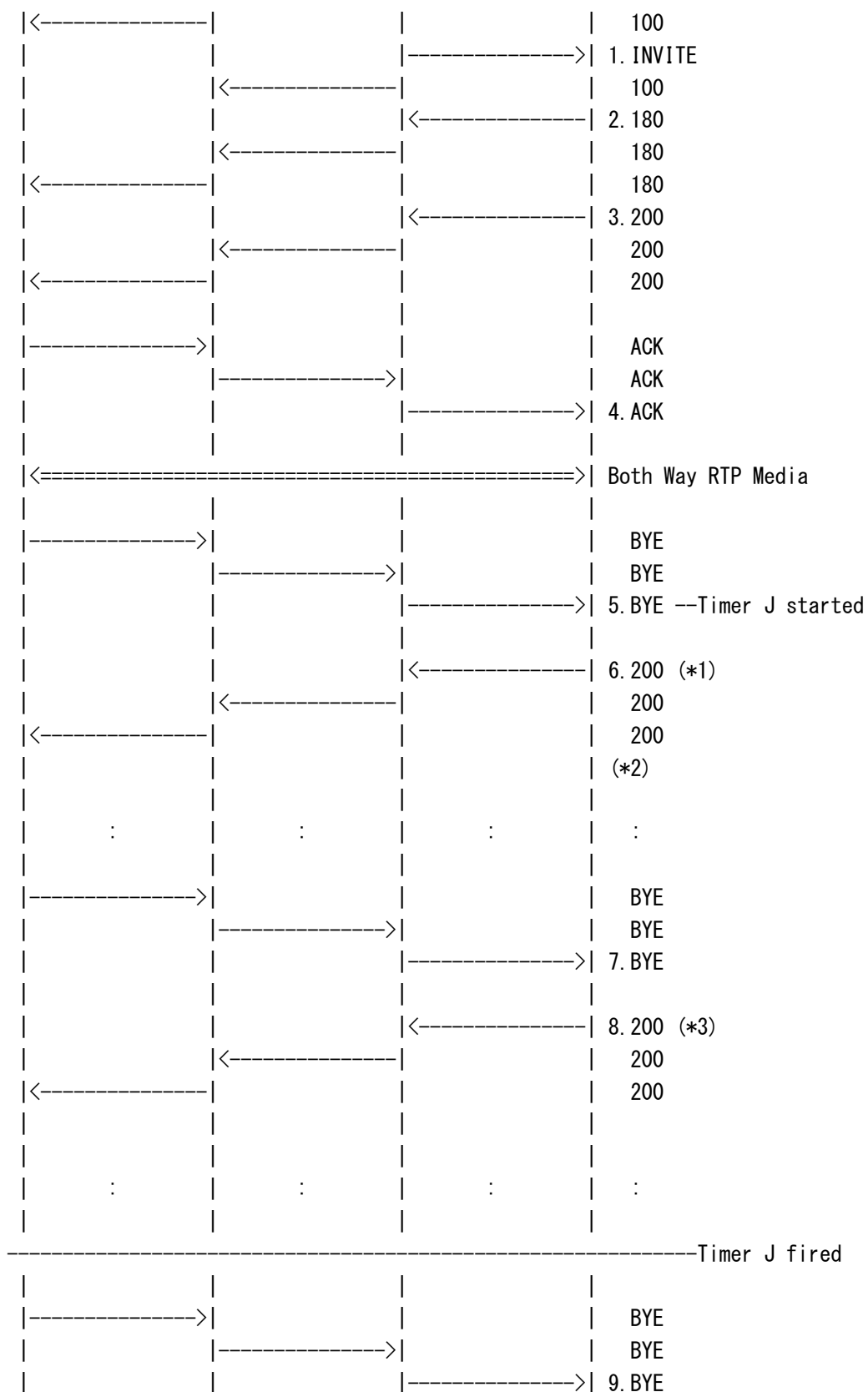
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

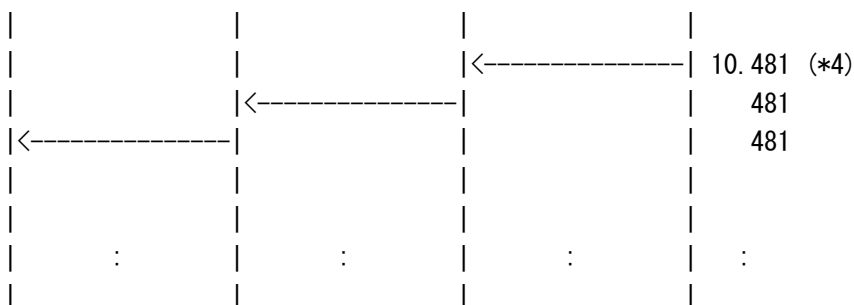
#### [PROCEDURE]

Note: This sequence is an example.

In some cases, the number of retransmission changes.







1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send BYE.-- Before Timer J      \* Timer J has started counting down...
6. Receive 200 OK. (\*1)
- (\*2)
7. Send BYE.-- every 2 seconds (Note: Timer J doesn't reset)
8. Receive 200 OK. (\*3)
9. Send BYE -- After Timer J terminates.
10. Receive 481 Call/Transaction Does Not Exist. (\*4)
- : repeat 9-10    every 2 seconds
- finish J + 30sec

#### [OBSERVABLE RESULTS]

- \*\* this scenario checks only timing (message format is not checked, except 1xx response)
- \*\* response(481 is used below) may be 300-699 response.

\*1:after 200 response from NUT.

Must not retransmit further 200 response. [RFC3261-17-83, 84, 85, 86]

\*2:200 response from NUT.(BYE is sent every 2 seconds)

Must send 200 response. [RFC3261-17-80][RFC3261-17-81][RFC3261-17-82]

Must send response before Timer J(=64\*T1 sec.) fires. [RFC3261-17-83]

\*3:after Timer J fires.(BYE is sent every 2 seconds)



Must send 481 response. [RFC3261-17-85][RFC3261-17-86]

#### [REFERENCE]

[RFC3261-17-83, 84, 85, 86]

#### 17.2.2 Non-INVITE Server Transaction

When the server transaction enters the "Completed" state, it MUST set Timer J to fire in  $64 \cdot T1$  seconds for unreliable transports, and zero seconds for reliable transports. While in the "Completed" state, the server transaction MUST pass the final response to the transport layer for retransmission whenever a retransmission of the request is received. Any other final responses passed by the TU to the server transaction MUST be discarded while in the "Completed" state. The server transaction remains in this state until Timer J fires, at which point it MUST transition to the "Terminated" state.

The server transaction MUST be destroyed the instant it enters the "Terminated" state.

### 4.3.15 UA-4-2-1 - 487 to CANCEL for INVITE request after $64 \cdot T1$ fired

#### [NAME]

UA-4-2-1 - 487 to CANCEL for INVITE request after  $64 \cdot T1$  fired

#### [PURPOSE]

Verify that a NUT properly sends a 487 response to previous CANCEL request for INVITE request after  $64 \cdot T1$  is fired.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

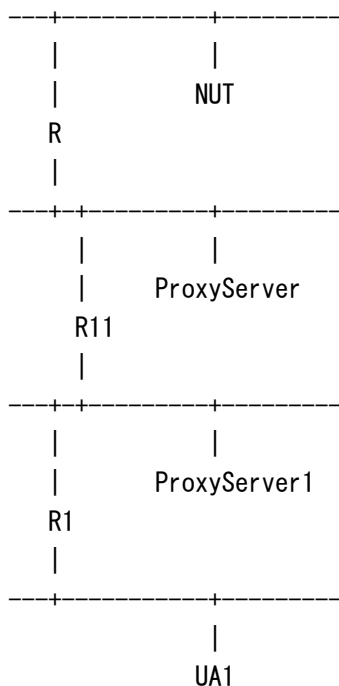
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

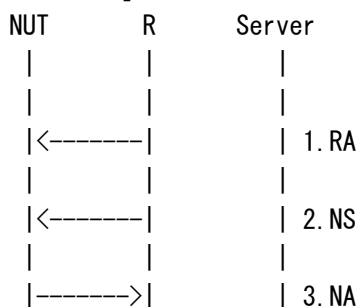
#### [TOPOLOGY]

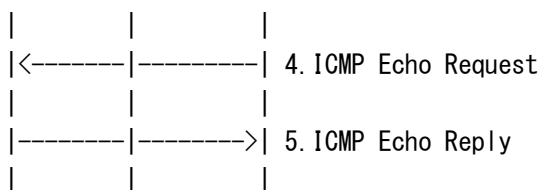


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

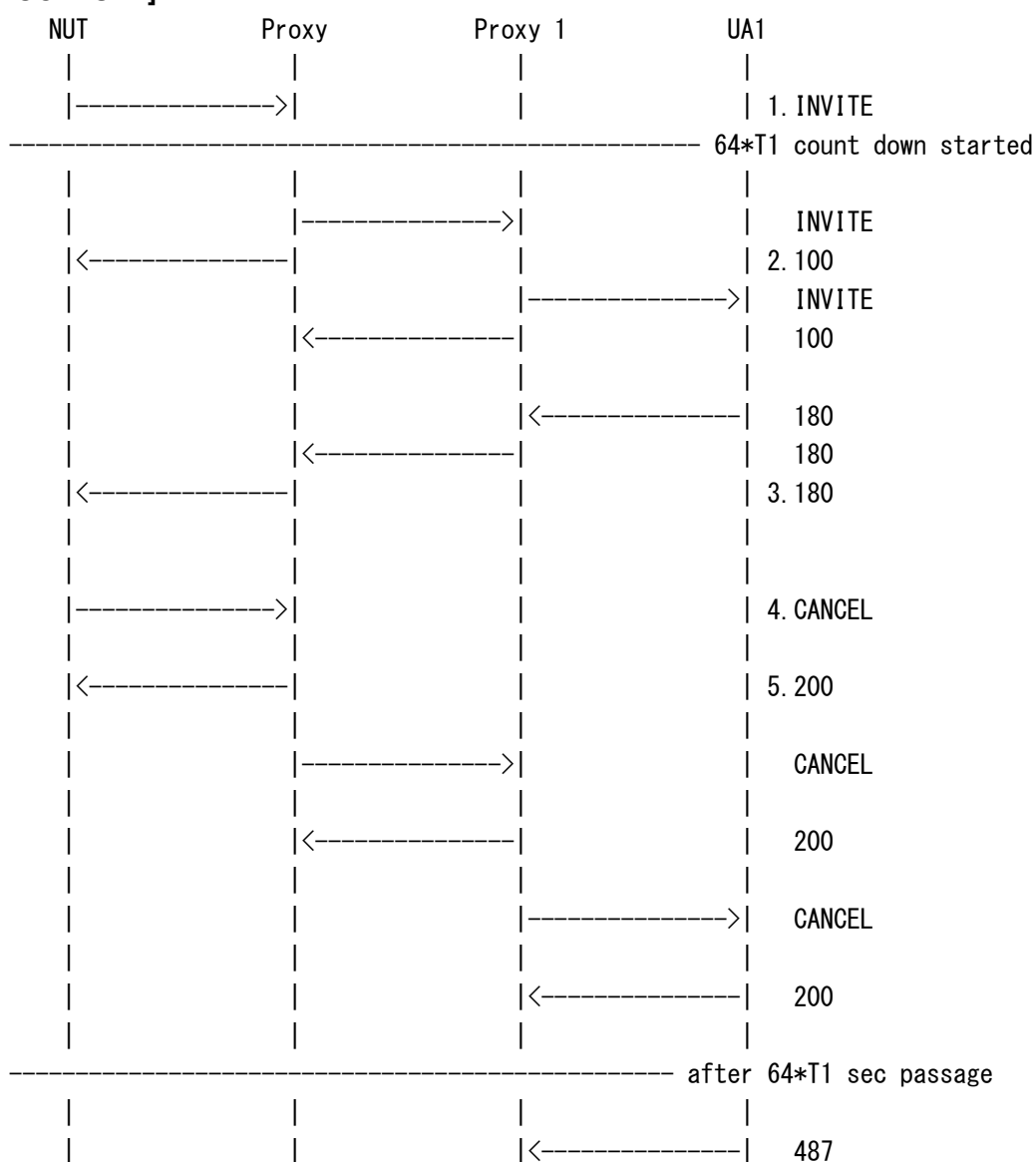
#### [INITIALIZATION]



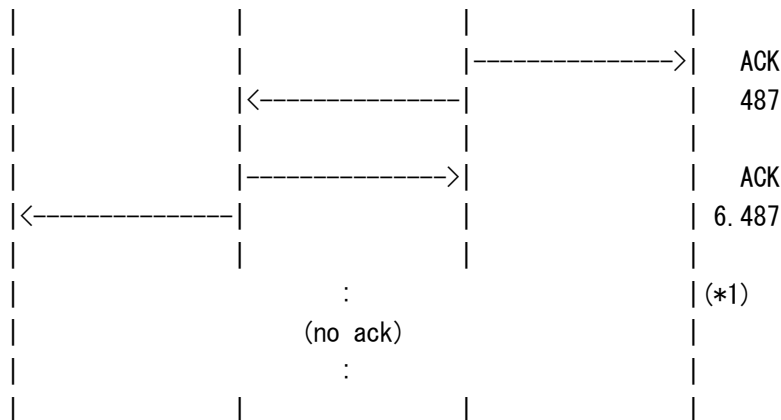


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Receive INVITE.
2. Send 100 Trying.
3. Send 180 Ringing.
4. Receive CANCEL.
5. Send 200 OK.
6. Send 487 Request Terminated.
- (\*1)

#### [OBSERVABLE RESULTS]

\*\* this scenario checks only timing (message format is not checked)

\*1:after 487 response from Proxy.

\* Should not send ACK request. [RFC3261-9-12][RFC3261-9-13]

#### [REFERENCE]

[RFC3261-9-12, 13]

9.1 Client Behavior

[RFC3261 Page 54 Paragraph 8]

Note that both the transaction corresponding to the original request and the CANCEL transaction will complete independently. However, a UAC canceling a request cannot rely on receiving a 487 (Request Terminated) response for the original request, as an RFC 2543-compliant UAS will not generate such a response. If there is no final response for the original request in  $64 \cdot T_1$  seconds ( $T_1$  is defined in Section 17.1.1.1), the client SHOULD then consider the original transaction cancelled and SHOULD destroy the client transaction handling the original request.

#### 4.3.16 UA-4-2-5 - Time in Expires in INVITE is up (UAS)

##### [NAME]

UA-4-2-5 - Time in Expires in INVITE is up (UAS)

##### [PURPOSE]

Verify that a NUT (UAS) properly processes when the time in an Expire parameter in an INVITE request is up.

##### [REQUIREMENT]

Only when A UA supports an Expires header field.

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

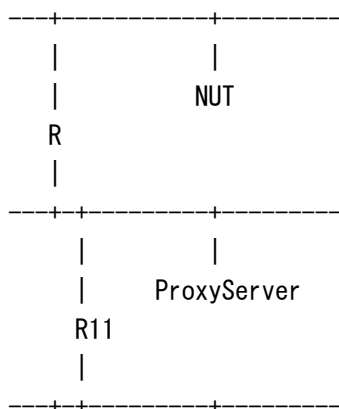
##### [PARAMETER]

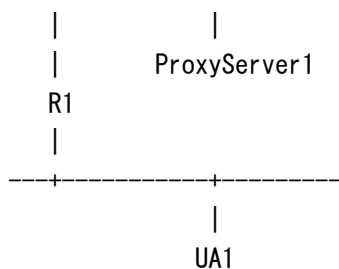
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

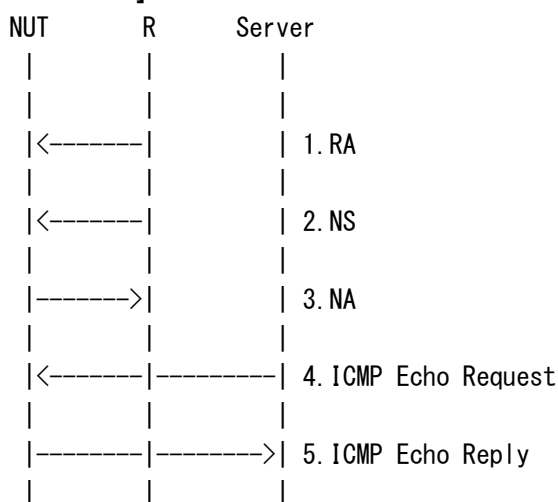




#### [CONFIGURATION for NUT]

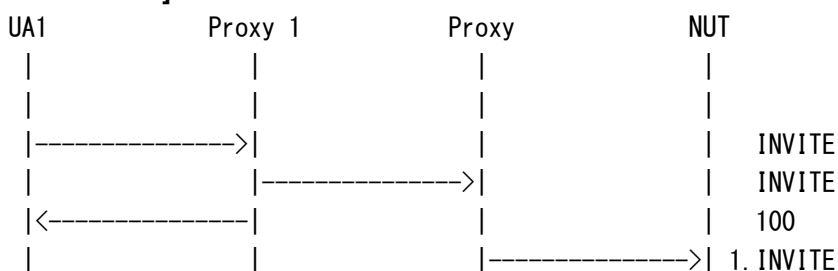
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

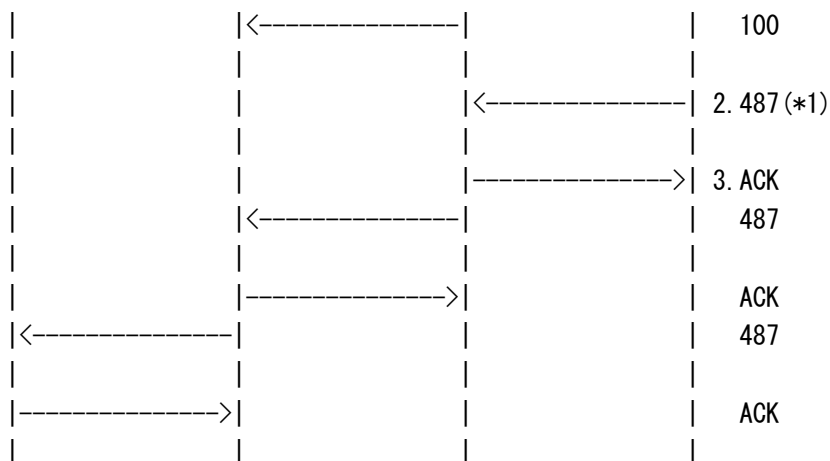
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE (including Expires header field(ex. "Expires: 2")).
2. Receive 487 Request Terminated. (\*1)
3. Send ACK.

#### [OBSERVABLE RESULTS]

\*1:487 response from NUT.

\* Should send 487 response. [RFC3261-13-27]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "487". [RFC3261-13-27]
- Header fields:  
See generic\_response
  - \* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

#### [REFERENCE]

[RFC3261-13-27]  
13.3.1 Processing of the INVITE  
[RFC3261 Page 83 Paragraph 5]



1. If the request is an INVITE that contains an Expires header field field, the UAS core sets a timer for the number of seconds indicated in the header field value. When the timer fires, the invitation is considered to be expired. If the invitation expires before the UAS has generated a final response, a 487 (Request Terminated) response SHOULD be generated.

#### 4.3.17 UA-4-2-6 - BYE for no response to retransmitted 200 (UAS)

##### [NAME]

UA-4-2-6 - BYE for no response to retransmitted 200 (UAS)

##### [PURPOSE]

Verify that a NUT (UAS) sends a BYE request when the other UA (UAC) doesn't reply any response to retransmitted 200 responses..

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

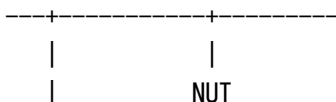
##### [PARAMETER]

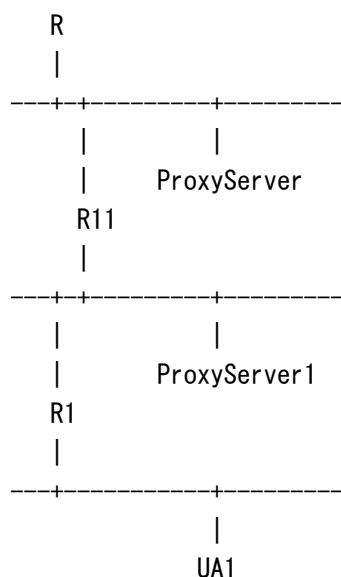
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

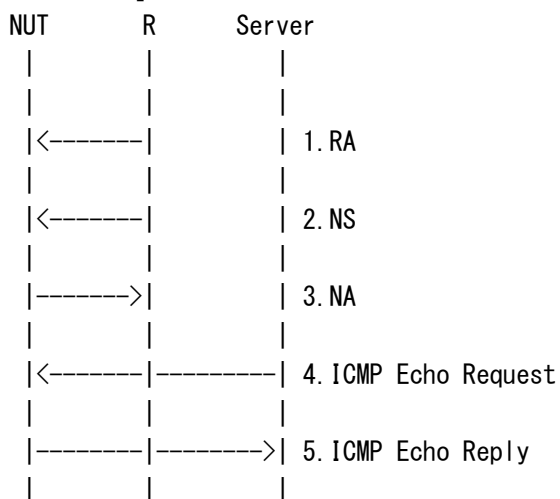




#### [CONFIGURATION for NUT]

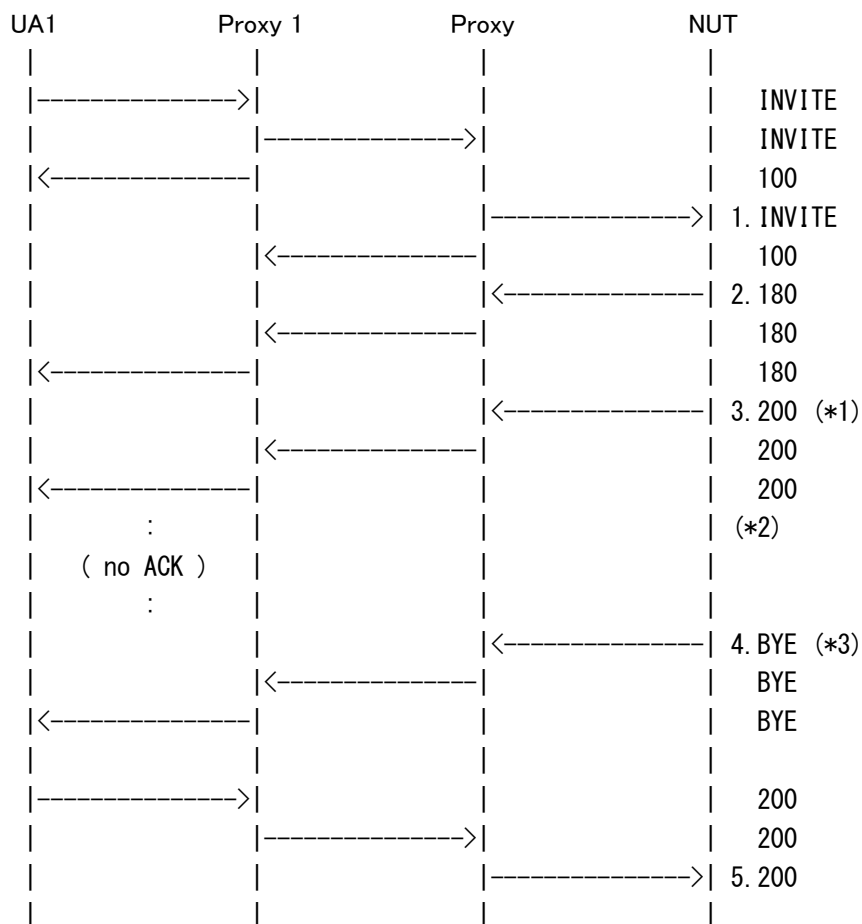
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]



1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK. (\*1)
- (\*2)
4. Receive BYE. (\*3)
5. Send 200 OK.

## [OBSERVABLE RESULTS]

\*1:200 response from NUT.

\* Must be retransmitted after Timer A(= T1 sec.) fired. [RFC3261-13-39]

\* Recommended not to be retransmitted with intervals that is shorter than 500msec. [RFC3261-17-12]

\*2:200 response from NUT.



\* Must be retransmitted after  $2 * T1$  sec. [RFC3261-17-8][RFC3261-17-9]

\*3:BYE request from NUT.

\* Should send BYE request. [RFC3261-13-39]

#### [REFERENCE]

[RFC3261-13-39, ]

13.3.1.4 The INVITE is Accepted

[RFC3261 Page 86 Paragraph 2]

Once the response has been constructed, it is passed to the INVITE server transaction. Note, however, that the INVITE server transaction will be destroyed as soon as it receives this final response and passes it to the transport. Therefore, it is necessary to periodically pass the response directly to the transport until the ACK arrives. The 2xx response is passed to the transport with an interval that starts at  $T1$  seconds and doubles for each retransmission until it reaches  $T2$  seconds ( $T1$  and  $T2$  are defined in Section 17). Response retransmissions cease when an ACK request for the response is received. This is independent of whatever transport protocols are used to send the response.

Since 2xx is retransmitted end-to-end, there may be hops between UAS and UAC that are UDP. To ensure reliable delivery across these hops, the response is retransmitted periodically even if the transport at the UAS is reliable.

If the server retransmits the 2xx response for  $64 * T1$  seconds without receiving an ACK, the dialog is confirmed, but the session SHOULD be terminated. This is accomplished with a BYE, as described in Section 15.

[RFC3261-15-4]

15 Terminating a Session

[RFC3261 Page 90 Paragraph 1]

However, the callee's UA MUST NOT send a BYE on a confirmed dialog until it has received an ACK for its 2xx response or until the server transaction times out. If no SIP extensions have defined other application layer states associated with the dialog, the BYE also



terminates the dialog.

#### 4.3.18 UA-4-2-7 - No Record-Route in negative replies (caller)

##### [NAME]

UA-4-2-7 - No Record-Route in negative replies (caller)

##### [PURPOSE]

Verify that a NUT (caller) properly receives a negative reply without a Record-Route header field.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

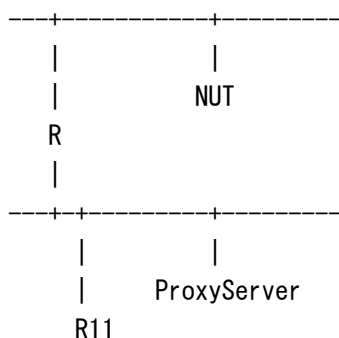
##### [PARAMETER]

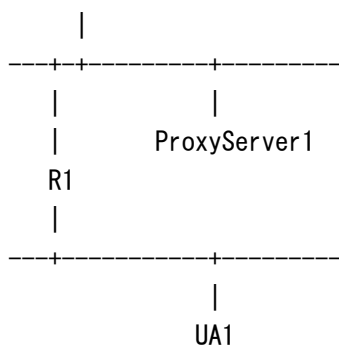
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

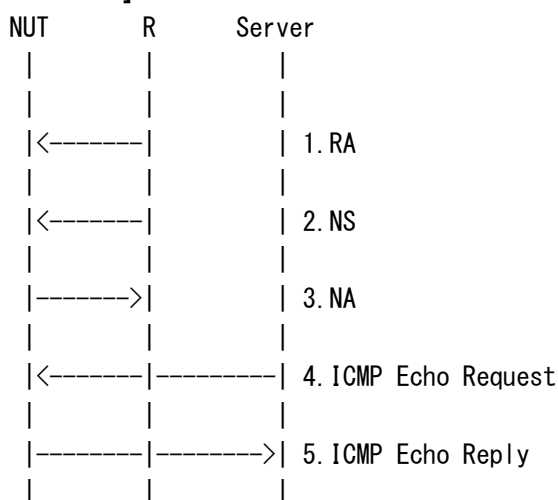




### [CONFIGURATION for NUT]

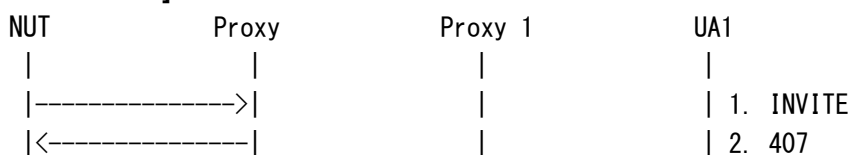
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

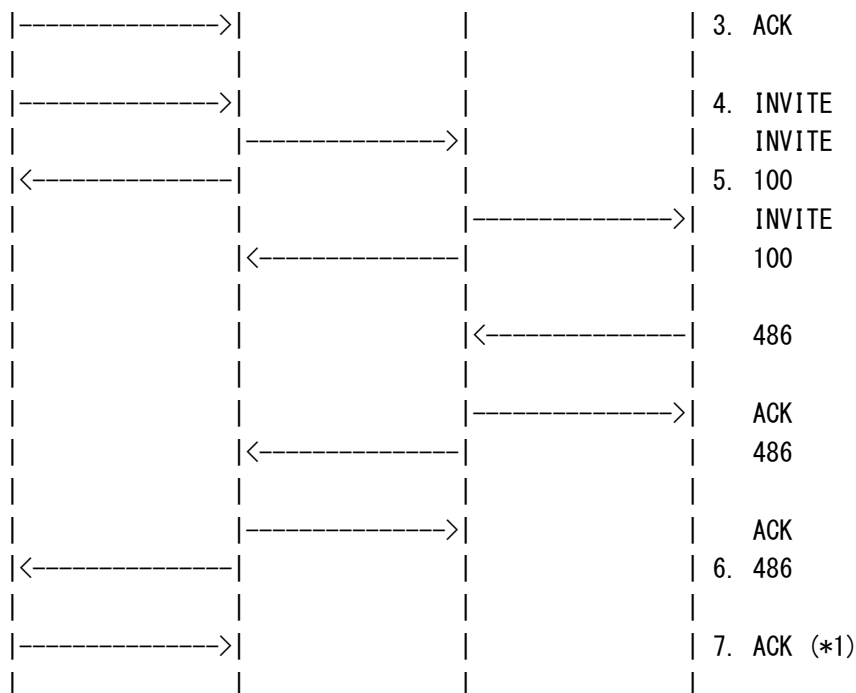
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 486 Busy Here.
7. Receive ACK. (\*1)

#### ===== Message example =====

##### 6. 486 Busy Here Proxy -> NUT

SIP/2.0 486 Busy Here  
 Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
 ;received=3ffe:501:ffff:5::X  
 Record-Route: <sip:ss.under.test.com;lr>,  
 <sip:ss1.atlanta.example.com;lr>  
 From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
 To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
 Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
 CSeq: 2 INVITE  
 Content-Length: 0

\* including Record-Route header field in non-2xx response.

#### [OBSERVABLE RESULTS]



\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

\* Route  
Must not exist. (It is invalid to set Route header field  
based on Record-Route header field in non-2xx response) [RFC3261-12-24]

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

## [REFERENCE]

[RFC3261-12-24]

### 12.1.1 UAS behavior

The route set MUST be set to the list of URIs in the Record-Route header field from the request, taken in order and preserving all URI parameters. If no Record-Route header field is present in the request, the route set MUST be set to the empty set. This route set, even if empty, overrides any pre-existing route set for future requests in this dialog. The remote target MUST be set to the URI from the Contact header field of the request.

## 4.3.19 UA-4-2-8 - No Record-Route in negative replies (callee)

### [NAME]

UA-4-2-8 - No Record-Route in negative replies (callee)

**[PURPOSE]**

Verify that a NUT (callee) properly sends a negative reply without a Record-Route header field.

**[REQUIREMENT]**

NONE

**[TARGET]**

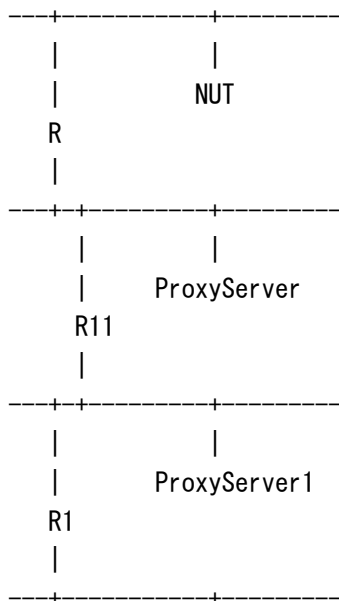
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

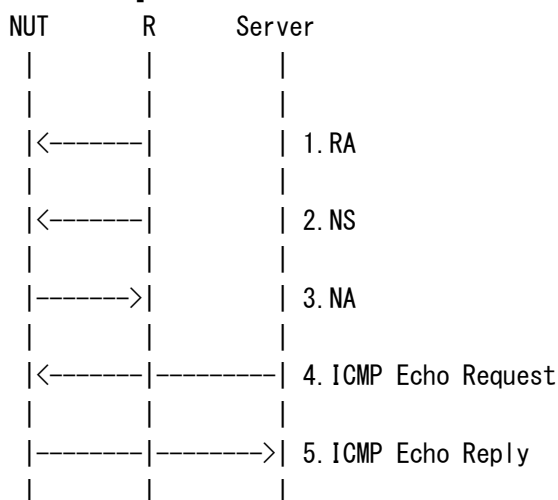
**[TOPOLOGY]**

UA1

### [CONFIGURATION for NUT]

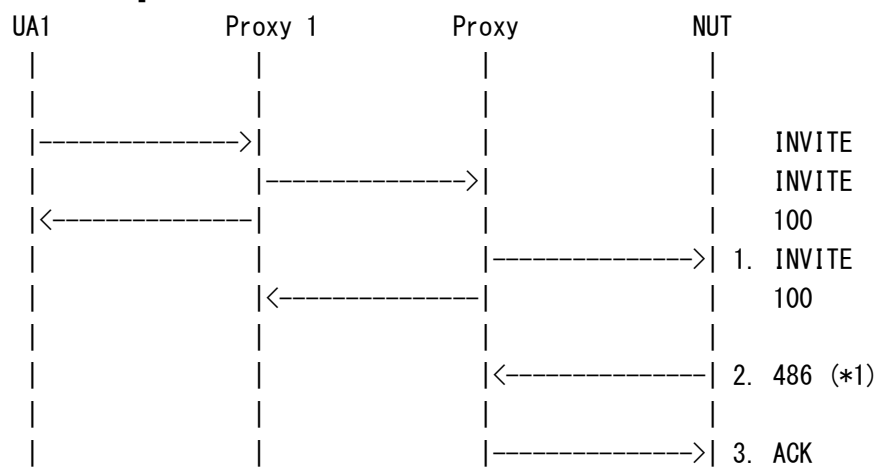
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

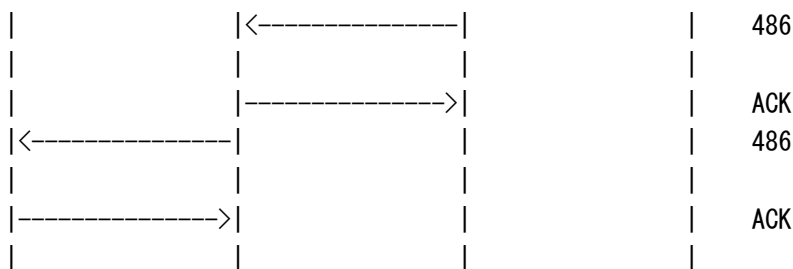
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 486 Busy Here. (any 4xx response is OK) (\*1)
3. Send ACK.

===== Message example =====

**1. INVITE Proxy -> NUT**

```
INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 151
```

```
v=0
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

**2. 486 Busy Here NUT -> Proxy**

```
SIP/2.0 486 Busy Here
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
```



;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:486 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must not exist. [RFC3261-12-10]

### [REFERENCE]

[RFC3261-12-10]

12.1.1 UAS Behavior

If no Record-Route header field is present in the request,  
the route set **MUST** be set to the empty set.



## 4.4 Mid-dialog control

### 4.4.1 UA-5-1-1 - Session with re-INVITE (Receiving re-INVITE for Hold) (Caller)

#### [NAME]

UA-5-1-1 - Session with re-INVITE (Receiving re-INVITE for Hold) (Caller)

#### [PURPOSE]

Verify that a NUT properly processing a re-INVITE for hold.

#### [REQUIREMENT]

Only when a NUT supports hold using re-INVITE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

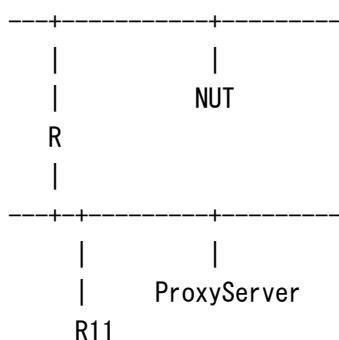
#### [PARAMETER]

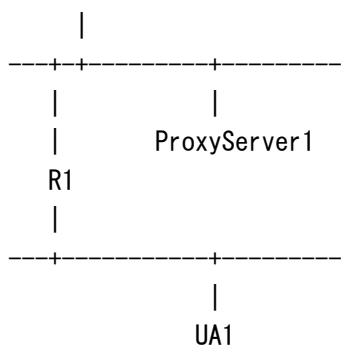
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

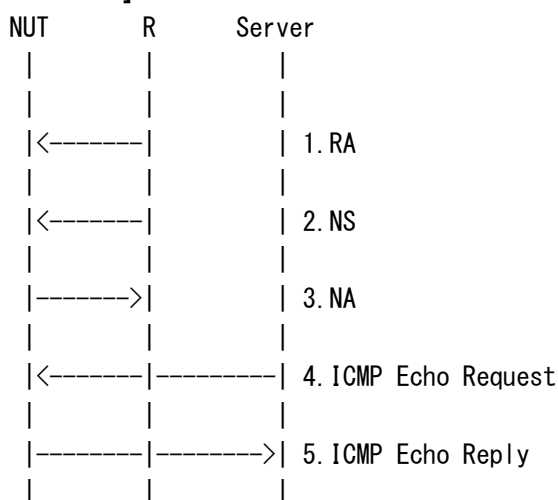




#### [CONFIGURATION for NUT]

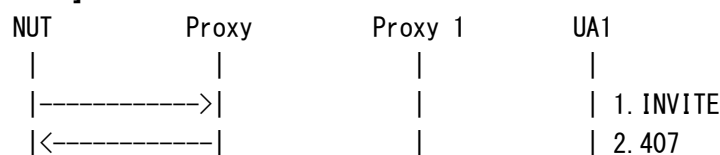
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

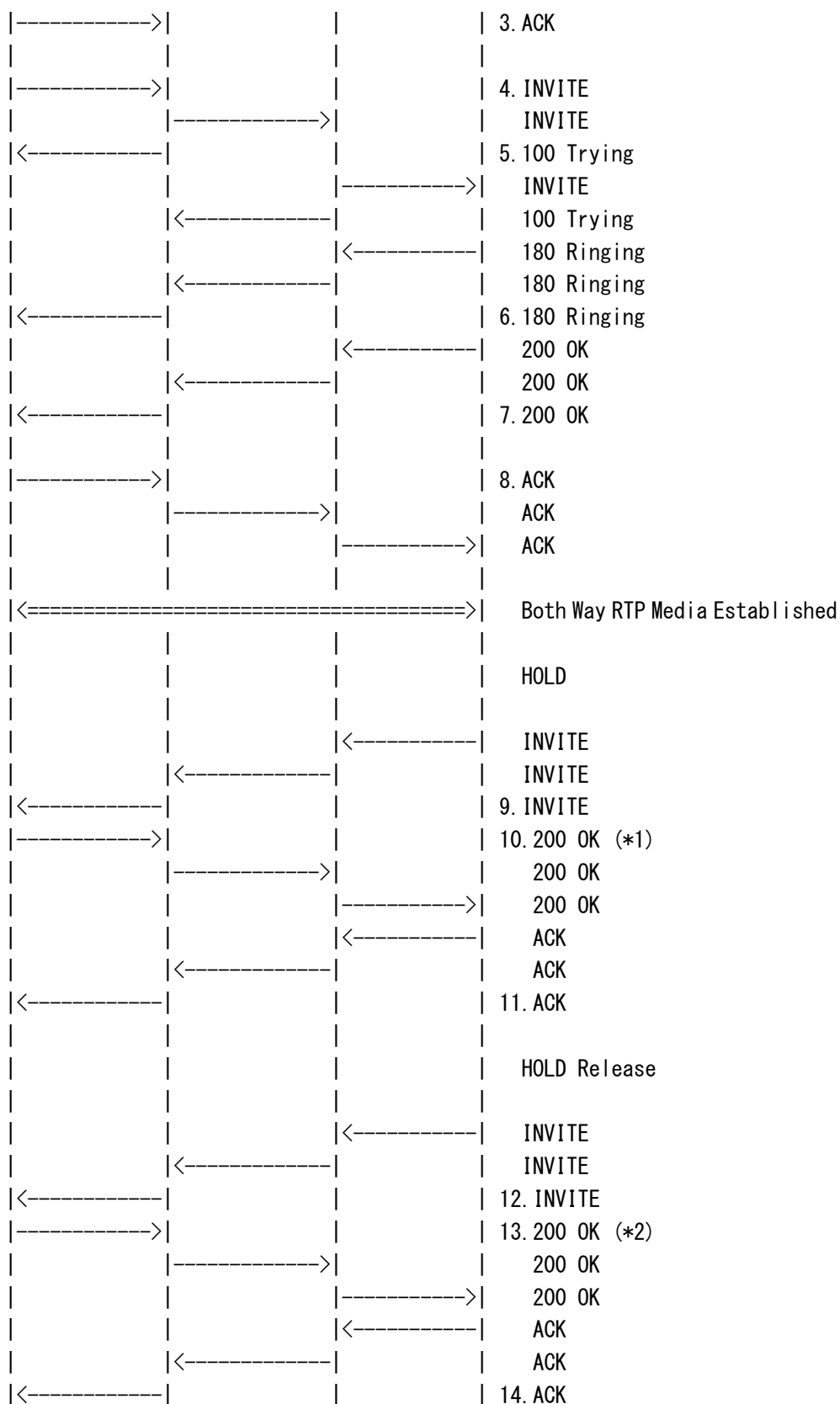
#### [INITIALIZATION]

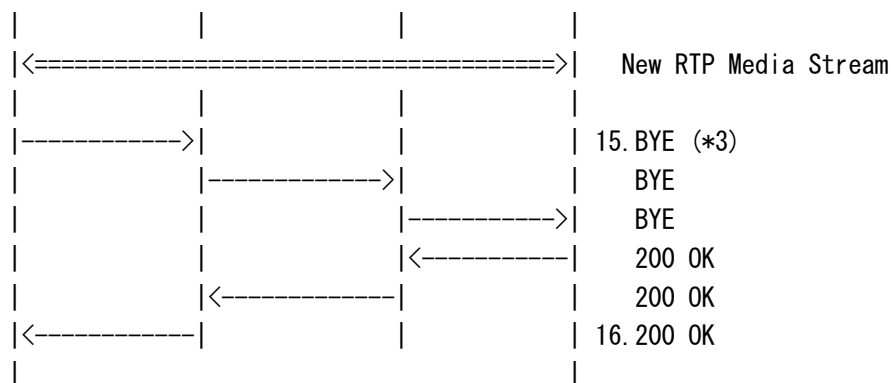


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Send INVITE.
10. Receive 200 OK. (\*1)
11. Send ACK.
12. Send INVITE.
13. Receive 200 OK. (\*2)
14. Send ACK.
15. Receive BYE. (\*3)
16. Send 200 OK.

#### ===== Message example =====

1. INVITE NUT -> Proxy

```

INVITE sip:UA1@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43
Max-Forwards: 70
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 2xTb9vxSit55XU7p8@under.test.com
CSeq: 1 INVITE
Contact: <sip:NUT@node.under.test.com>
Content-Type: application/sdp
Content-Length: 151
  
```



v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 2. 407 Proxy Authorization Required Proxy -> NUT

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="f84f1cec41e6cbe5aea9c8e88d359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

## 3. ACK NUT -> Proxy

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 ACK  
Content-Length: 0

## 4. INVITE NUT -> Proxy

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",



response="b51e504e73af54829e4f2bd7f8dc4654"  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### 5. 100 Trying Proxy -> NUT

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

#### 6. 180 Ringing Proxy -> NUT

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Length: 0

#### 7. 200 OK Proxy -> NUT



SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### 8. ACK NUT -> Proxy

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b7b  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

#### 9. INVITE Proxy -> NUT



INVITE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 15 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 149

v=0  
o=UA1 2890844527 2890844528 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=sendonly

10. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 15 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X





s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=recvonly

#### 11. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 15 ACK  
Content-Length: 0

#### 12. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 16 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 149

v=0  
o=UA1 2890844527 2890844529 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1



t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

13. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 16 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

14. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 16 ACK  
Content-Length: 0



#### 15. BYE NUT -> Proxy

BYE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bo4  
Route: <sip:ss.under.test.com;lr>,  
      <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 3 BYE  
Content-Length: 0

#### 16. 200 OK Proxy -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bo4  
      ;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 3 BYE  
Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- inside of a dialog  
See generic\_200-for-INVITE



\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

See generic\_200-for-INVITE

Status-Code: Must be "200".

- Header fields:

See generic\_response

- inside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]



Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

\*3:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "7.200" response. [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the To header field of "7.200" response. [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters. [RFC3261-12-48]

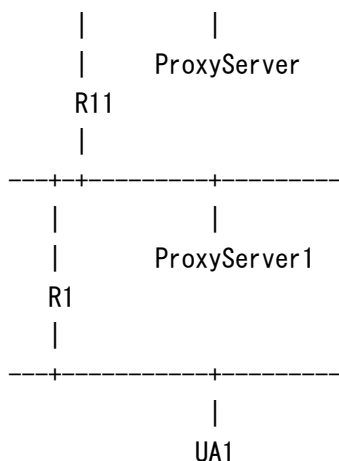
- Bodies:

See generic\_BYE

(Special mention matter)

Must not initiate a new INVITE transaction within a dialog while another

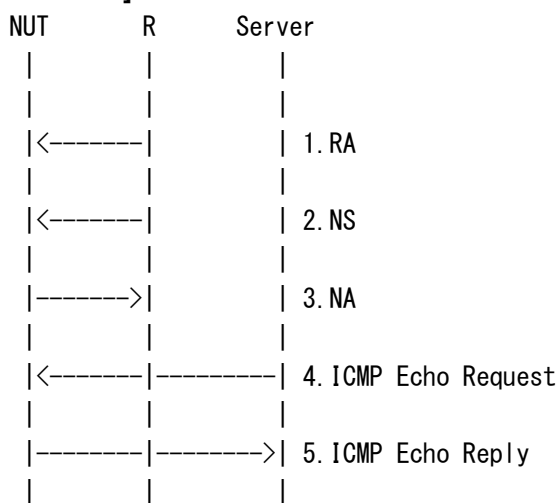




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

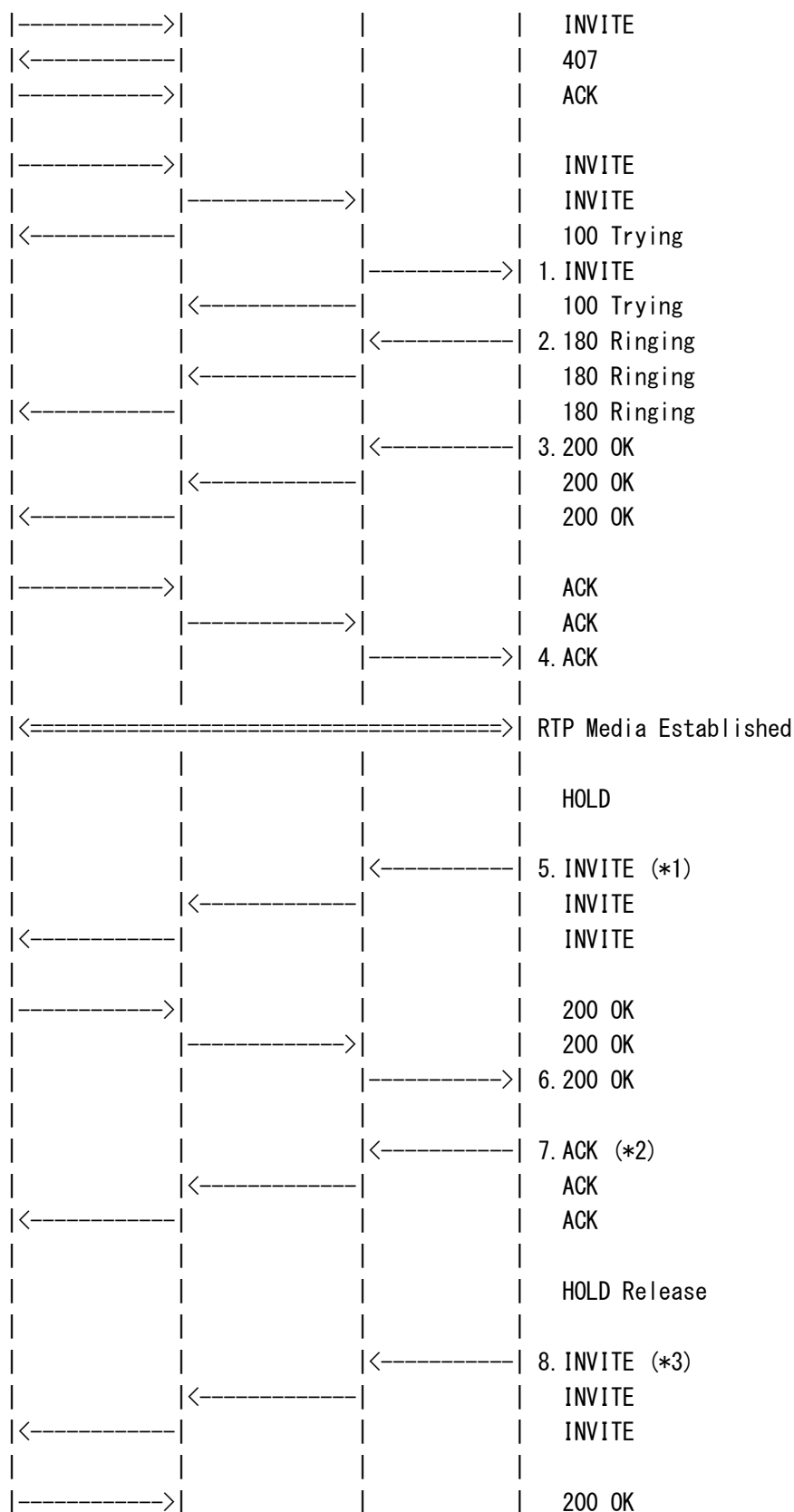
#### [INITIALIZATION]



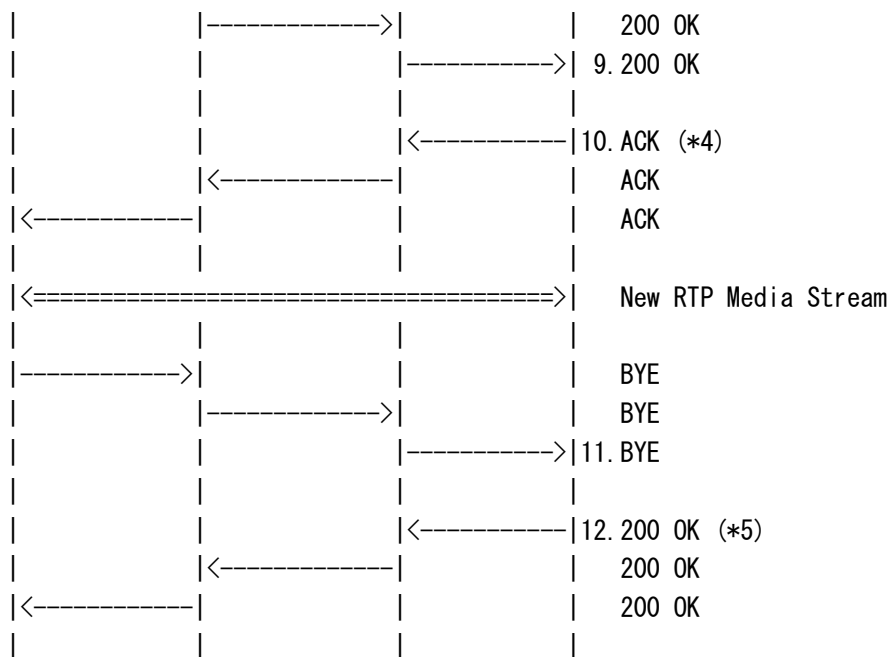
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]









1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Receive INVITE. (\*1)
6. Send 200 OK.
7. Receive ACK. (\*2)
8. Receive INVITE. (\*3)
9. Send 200 OK.
10. Receive ACK. (\*4)
11. Send BYE.
12. Receive 200 OK. (\*5)

#### ===== Message example =====

1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
Max-Forwards: 68
  
```



Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 2. 180 Ringing NUT -> Proxy

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Length: 0

## 3. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1



;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
              <sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=NUT 2890844527 2890844527 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### 4. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74b76  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 ACK  
Content-Length: 0

#### 5. INVITE NUT -> Proxy

INVITE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKlkld5l  
Max-Forwards: 70  
Route: <sip:ss.under.test.com;lr>,



<sip:ss1.atlanta.example.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 14 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 149

v=0  
o=NUT 2890844527 2890844528 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=sendonly

6. 200 OK Proxy -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKlkld5l  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 14 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=recvonly

7. ACK NUT -> Proxy



ACK sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKlkldcc  
Route: <sip:ss.under.test.com;lr>,  
      <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 14 ACK  
Content-Length: 0

8. INVITE NUT -> Proxy

INVITE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKlkbqc  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 15 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 149

v=0  
o=NUT 2890844527 2890844529 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

9. 200 OK Proxy -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKlklbqc  
      ;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 15 INVITE



Contact: <sip:UA1@client.atlanta.example.com>  
Allow: REGISTER,INVITE,CANCEL,ACK,PRACK,UPDATE,BYE  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

10. ACK NUT -> Proxy

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKklbqc  
Route: <sip:ss.under.test.com;lr>,  
      <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 15 ACK  
Content-Length: 0

11. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
      ;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
      ;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 2 BYE  
Content-Length: 0

12. 200 OK NUT -> Proxy



SIP/2.0 200 OK

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1

;received=3ffe:501:ffff:50::50

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9

;received=3ffe:501:ffff:1::1

From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

To: NUT <sip:NUT@under.test.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com

CSeq: 2 BYE

Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_re-INVITE

Method: Must be "INVITE". [RFC3261 7.1]

Request-URI: Must be the URI of Contact in "1.INVITE". [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_re-INVITE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "2.200"  
response. [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]



route-param: Must contain the route set values(set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_re-INVITE

"a=" line: "a=sendonly" must be appended into SDP. [RFC3264-5-7]

"o=" line:

<version>: Must be incremented by one from <version> of previous SDP. [RFC3264-8-2]

"m=" line: At least, Must include the matching m line as SDP in the sending 3. 200 response included. [RFC3264-8-6]

\*2:ACK request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values(set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*3:INVITE request from NUT.

As a SIP Message,





See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_re-INVITE

Method: Must be "INVITE". [RFC3261 7.1]

Request-URI: Must be the URI of Contact in "1.INVITE". [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_re-INVITE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "2.200" response. [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values (set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_request

See generic\_re-INVITE

"a=" line: Either include an "a=sendrecv" attribute, or omit it. [RFC3264 5.1]

"o=" line:

<version>: Must be incremented by one from <version> of previous SDP. [RFC3264-8-2]

"m=" line: At least, Must include the matching m line as SDP in the sending 3. 200 response included. [RFC3264-8-6]

(tester checks only the number of "m=" line.) [RFC3264-8-6,7]

\*4:ACK request from NUT.

As a SIP Message,



See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values(set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*5:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "200".

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter



contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values(set by Record-Route header field in BYE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_response

See generic\_SDP

(Special mention matter)

automated generation of re-INVITE or BYE is not recommended. [RFC3261-14-1]

Must not initiate a new INVITE transaction within a dialog while another INVITE transaction is in progress in either direction. [RFC3261-14-4]

#### [REFERENCE]

Sequence from RFC3665 Section 3.7.

### 4.4.3 UA-5-2-1 - Receipt of re-INVITE before sending the final response to the first INVITE

#### [NAME]

UA-5-2-1 – Receipt of re-INVITE before sending the final response to the first INVITE

#### [PURPOSE]

Verify that a NUT properly receives a re-INVITE before sending the final response to the first INVITE.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

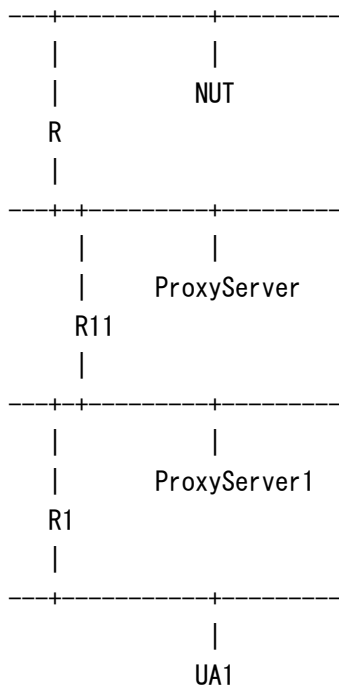
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com

Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

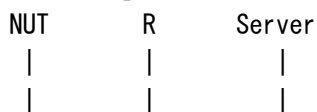
#### [TOPOLOGY]

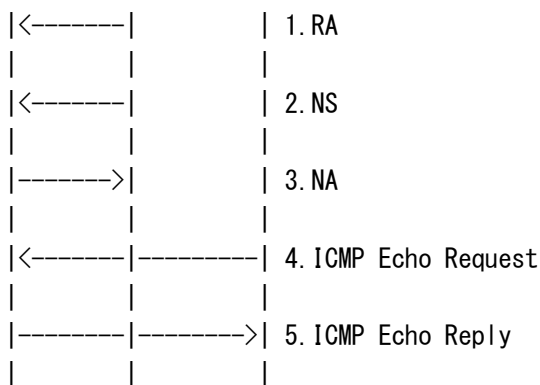


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

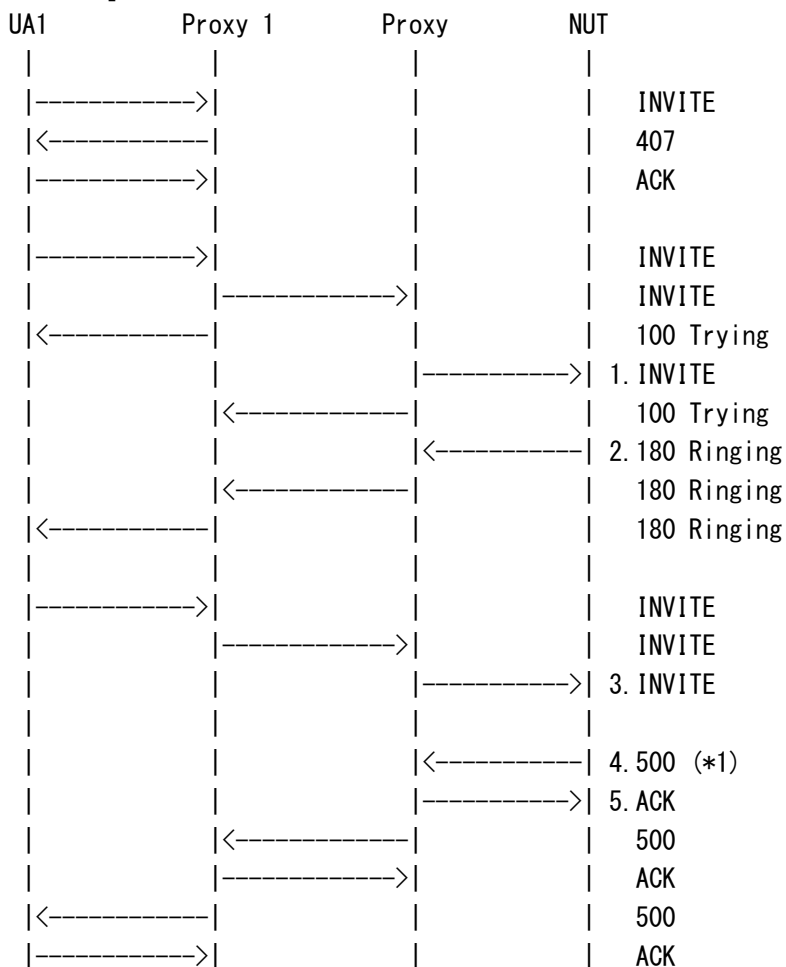
#### [INITIALIZATION]

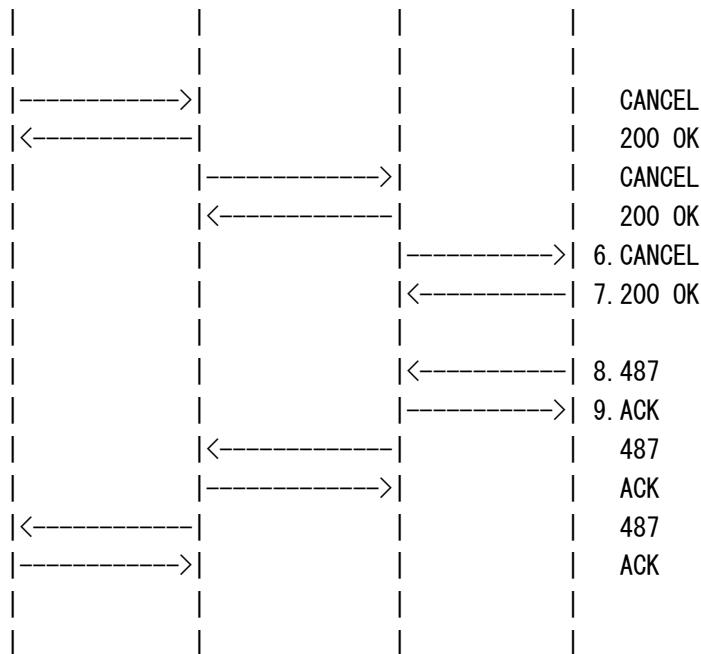




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Send INVITE.
4. Receive 500 Server Internal Error. (\*1)
5. Send ACK.
6. Send CANCEL.
7. Receive 200 OK.
8. Receive 487 Request Terminated.
9. Send ACK.

===== Message example =====

**1. INVITE Proxy -> NUT**

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>, <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com
CSeq: 1 INVITE
  
```



Contact: <sip:UA1@client.atlanta.example.com>

Content-Type: application/sdp

Content-Length: 151

v=0

o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1

s=-

c=IN IP6 3ffe:501:ffff:1::1

t=0 0

m=audio 49172 RTP/AVP 0

a=rtpmap:0 PCMU/8000

## 2. 180 Ringing NUT -> Proxy

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1

;received=3ffe:501:ffff:50::50

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9

;received=3ffe:501:ffff:1::1

Record-Route: <sip:ss.under.test.com;lr>, <sip:ss1.atlanta.example.com;lr>

From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

To: NUT <sip:NUT@under.test.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com

CSeq: 1 INVITE

Contact: <sip:NUT@node.under.test.com>

Content-Length: 0

## 3. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.11

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.11

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf10

;received=3ffe:501:ffff:1::1

Max-Forwards: 68

Record-Route: <sip:ss.under.test.com;lr>, <sip:ss1.atlanta.example.com;lr>

From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

To: NUT <sip:NUT@under.test.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com

CSeq: 2 INVITE



Contact: <sip:UA1@client.atlanta.example.com>

Content-Type: application/sdp

Content-Length: 163

v=0

o=UA1 2890844526 2890844527 IN IP6 3ffe:501:ffff:1::1

s=-

c=IN IP6 3ffe:501:ffff:1::1

t=0 0

m=audio 49172 RTP/AVP 0

a=rtpmap:0 PCMU/8000

a=sendonly

#### 4. 500 Server Internal Error NUT -> Proxy

SIP/2.0 500 Server Internal Error

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.11

;received=3ffe:501:ffff:50::50

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.11

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf10

;received=3ffe:501:ffff:1::1

Record-Route: <sip:ss.under.test.com;lr>, <sip:ss1.atlanta.example.com;lr>

From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

To: NUT <sip:NUT@under.test.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com

CSeq: 2 INVITE

Retry-After: 10

Content-Length: 0

#### 5. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.11

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.11

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf10

;received=3ffe:501:ffff:1::1

Max-Forwards: 68

Record-Route: <sip:ss.under.test.com;lr>, <sip:ss1.atlanta.example.com;lr>

From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

To: NUT <sip:NUT@under.test.com>;tag=314159

Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com





CSeq: 2 ACK  
Content-Length: 0

#### **[OBSERVABLE RESULTS]**

\*1:500 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "500". [RFC3261-14-10]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Retry-After  
Must exist. [RFC3261-14-11]  
detla-seconds: Must be between 0 and 10. [RFC3261-14-11]

#### **[REFERENCE]**

[RFC3261-14-10, 11]  
14.2 UAS Behavior

A UAS that receives a second INVITE before it sends the final response to a first INVITE with a lower CSeq sequence number on the same dialog MUST return a 500 (Server Internal Error) response to the second INVITE and MUST include a Retry-After header field with a randomly chosen value of between 0 and 10 seconds.

#### **4.4.4 UA-5-2-2 - Receipt of re-INVITE before receiving the final response to another re-INVITE**

##### **[NAME]**

UA-5-2-2 – Receipt of re-INVITE before receiving the final response to another re-INVITE



### [PURPOSE]

Verify that a NUT properly receives a re-INVITE before receiving the final response to another INVITE.

### [REQUIREMENT]

Only when a NUT supports hold using re-INVITE.

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

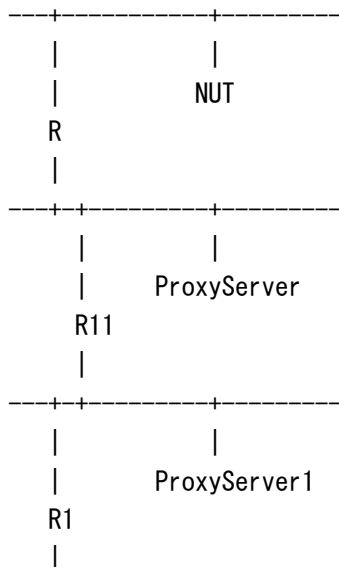
### [PARAMETER]

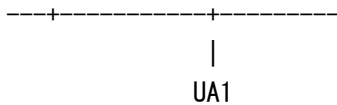
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]

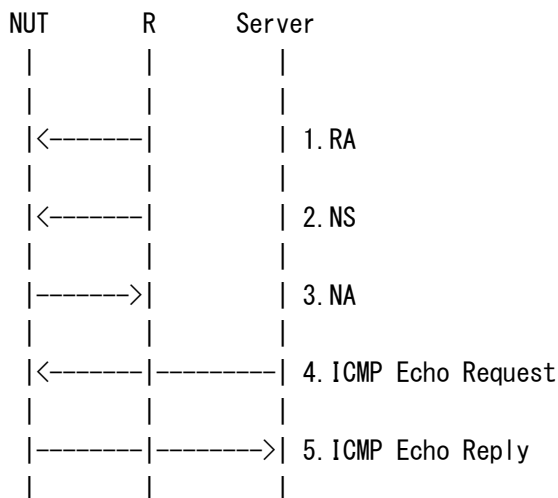




#### [CONFIGURATION for NUT]

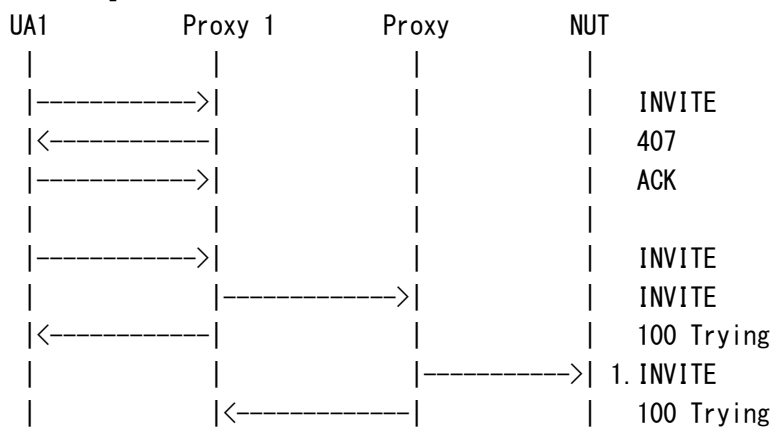
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

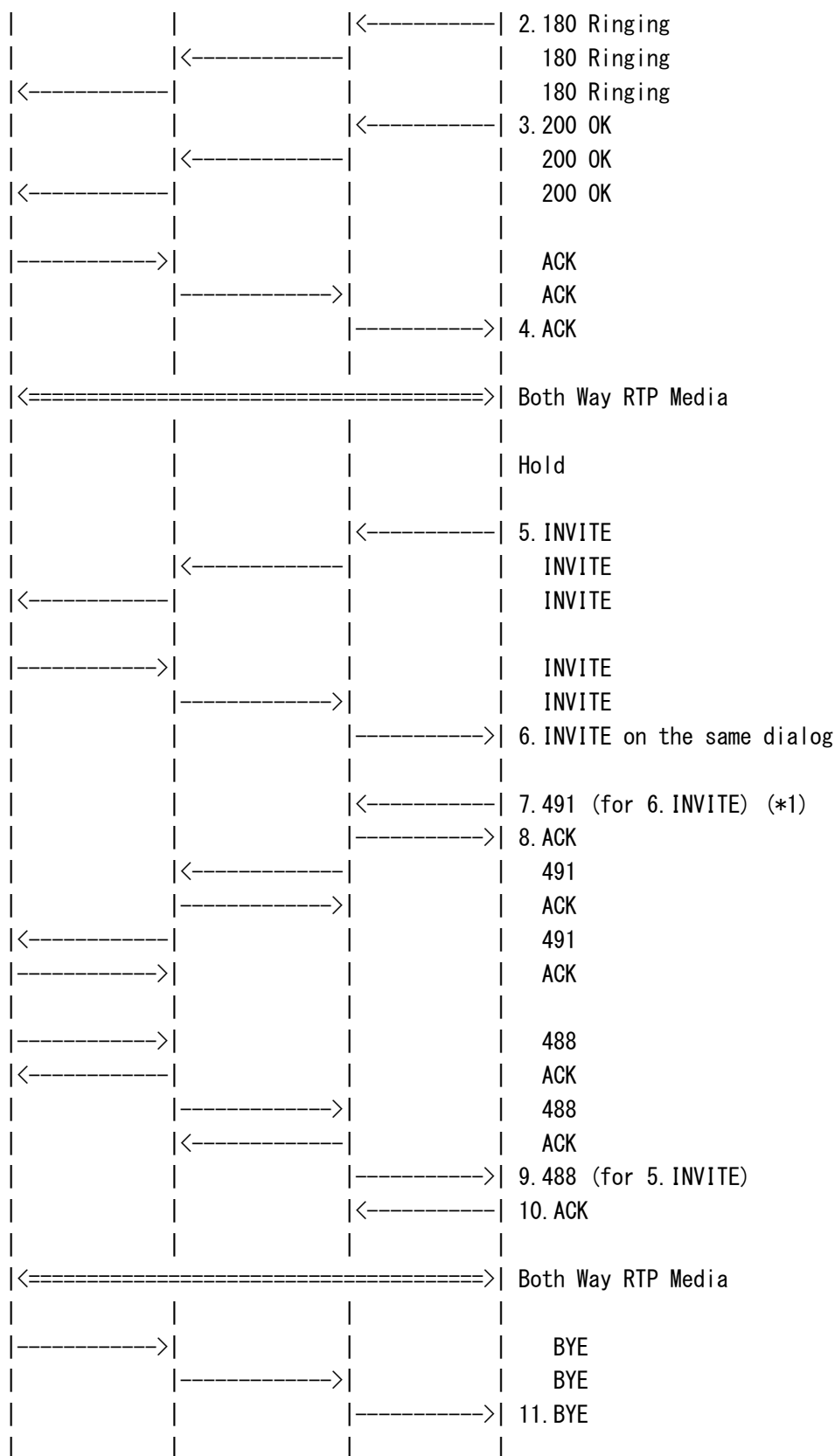
#### [INITIALIZATION]

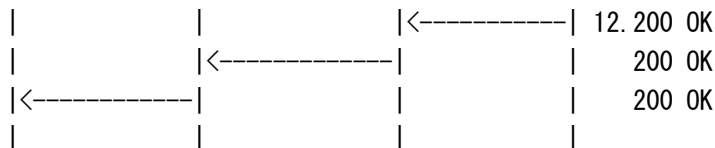


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Receive INVITE.
6. Send INVITE.
7. Receive 491 Request Pending. (\*1)
8. Send ACK.
9. Send 488.
10. Receive ACK.
11. Send BYE.
12. Receive 200 OK.

#### [OBSERVABLE RESULTS]

\*1:491 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "491". [RFC3261-14-12]

- Header fields:  
See generic\_response

- inside of a dialog

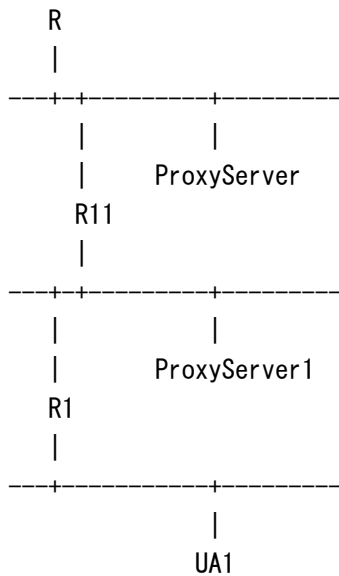
\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

- Bodies:  
See generic\_SDP

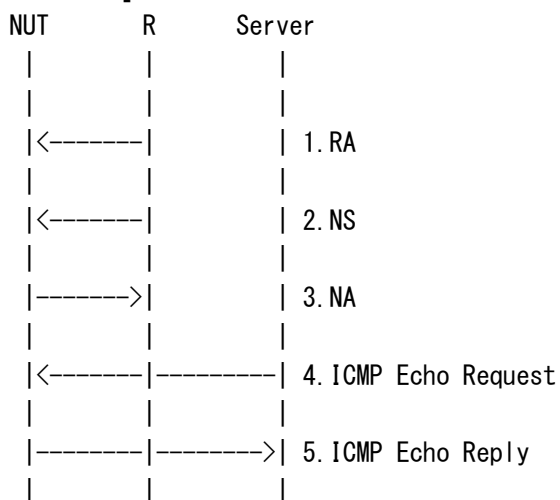




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

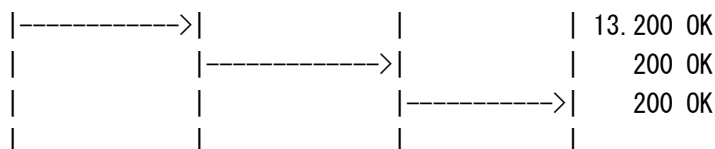
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.







1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Send INVITE.
10. Receive 488 Not Acceptable Here. (\*1)
11. Send ACK.
12. Send BYE.
13. Receive 200 OK.

===== Message example =====

**9. INVITE Proxy -> NUT**

INVITE sip:NUT@node.under.test.com SIP/2.0  
 (snip)  
 Content-Length: 169

v=0  
 o=UA1 2890844526 2890844527 IN IP6 3ffe:501:ffff:1::1  
 s=-  
 c=IN IP6 3ffe:501:ffff:1::1  
 t=0 0  
 m=audio 49172 RTP/AVP 999999999  
 a=rtpmap:999999999 PCMU/999999999

**[OBSERVABLE RESULTS]**

\*1:488 response from NUT.

As a SIP Message,  
 See generic\_message

As a SIP response,

- Status-Line:



See generic\_response

Status-Code: Must be "488". [RFC3261-14-34]

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Warning

Should exist. [RFC3261-14-15, 35]

- Bodies:

See generic\_SDP

#### [REFERENCE]

[RFC3261-14-15]

#### 14.2 UAS Behavior

If the new session description is not acceptable, the UAS can reject it by returning a 488 (Not Acceptable Here) response for the re-INVITE. This response SHOULD include a Warning header field.

[RFC3261-14-34, 35]

A UAS rejecting an offer contained in an INVITE SHOULD return a 488 (Not Acceptable Here) response. Such a response SHOULD include a Warning header field value explaining why the offer was rejected.

### 4.4.6 UA-5-2-4 - Unacceptable re-INVITE (Callee)

#### [NAME]

UA-5-2-4 - Unacceptable re-INVITE (Callee)

#### [PURPOSE]

Verify that a NUT (callee) properly processes an unacceptable re-INVITE.

#### [REQUIREMENT]

NONE

# **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

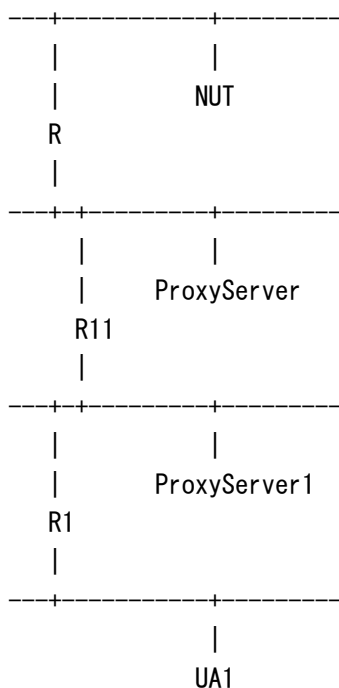
# **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

# **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# **[TOPOLOGY]**



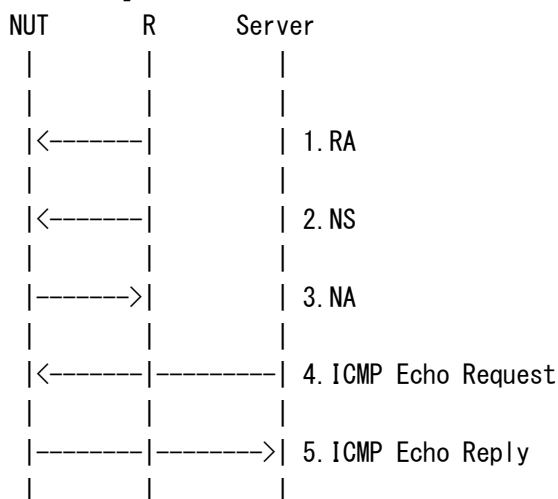
# **[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------



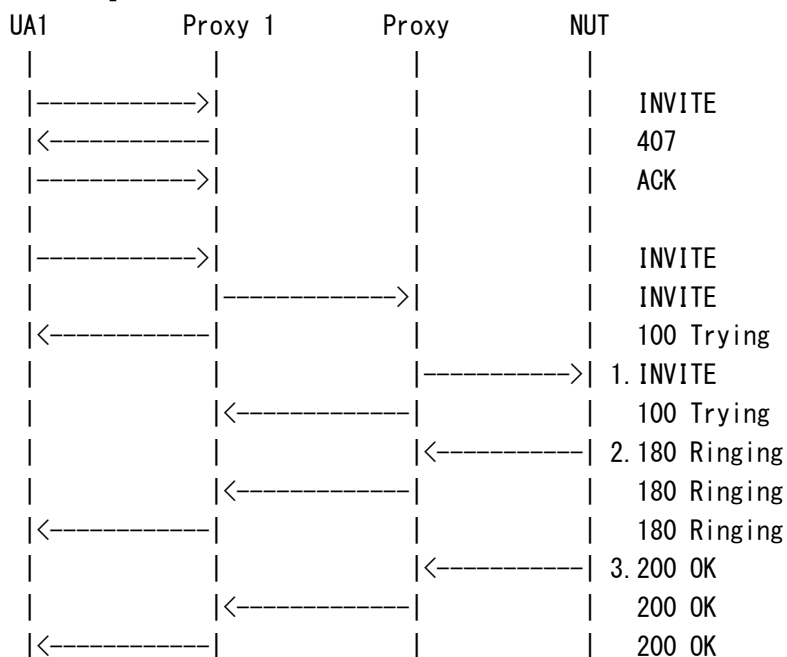
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

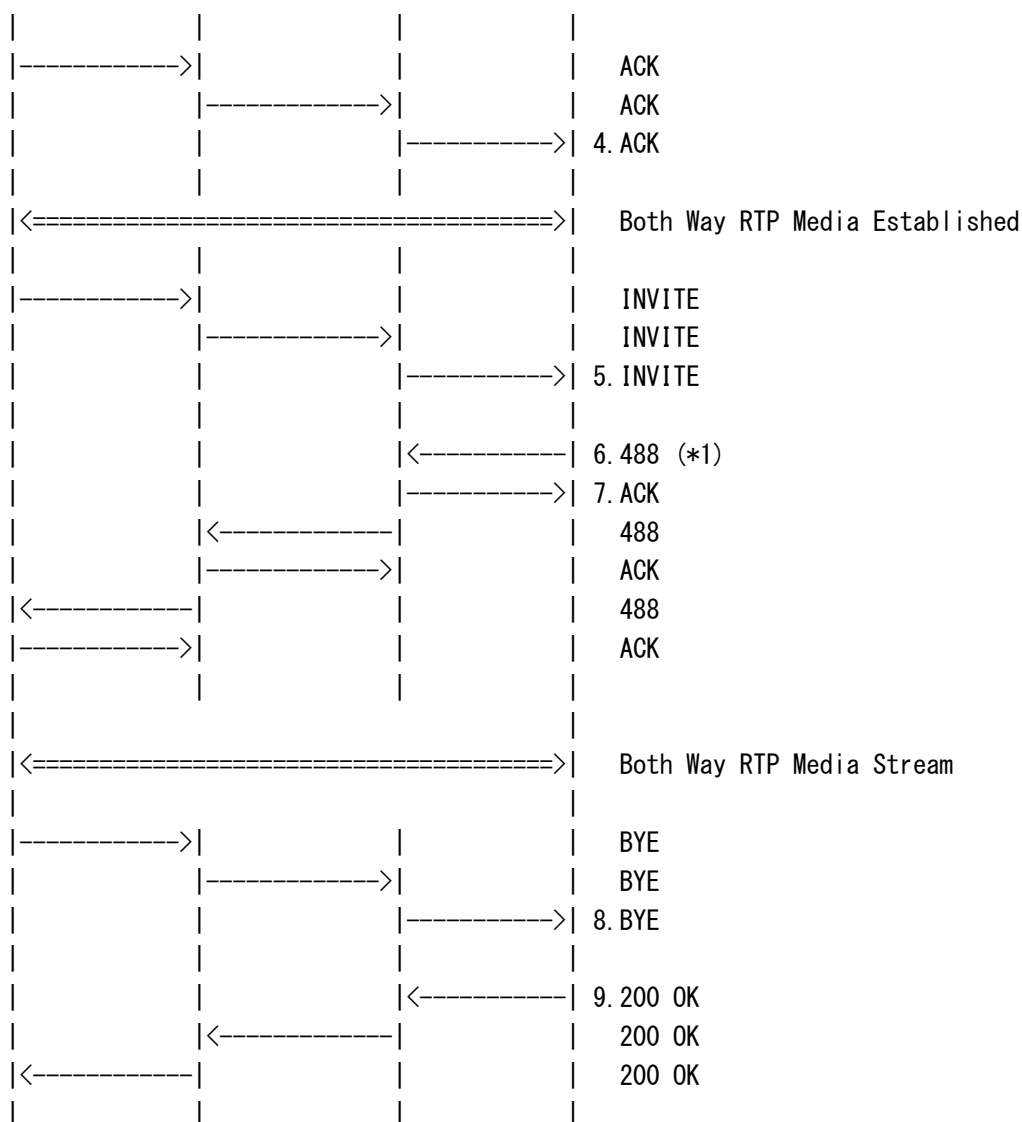
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send INVITE.
6. Receive 488 Not Acceptable Here. (\*1)
7. Send ACK.
8. Send BYE.
9. Receive 200 OK.

### 5. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0



(snip)

Content-Length: 169

v=0

o=UA1 2890844526 2890844527 IN IP6 3ffe:501:ffff:1::1

s=-

c=IN IP6 3ffe:501:ffff:1::1

t=0 0

m=audio 49172 RTP/AVP 999999999

a=rtpmap:999999999 PCMU/999999999

### [OBSERVABLE RESULTS]

\*1:488 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "488". [RFC3261-14-34]

- Header fields:  
See generic\_response

- inside of a dialog

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Warning  
Should exist. [RFC3261-14-15, 35]

- Bodies:  
See generic\_SDP

### [REFERENCE]

[RFC3261-14-15]

14.2 UAS Behavior



If the new session description is not acceptable, the UAS can reject it by returning a 488 (Not Acceptable Here) response for the re-INVITE. This response SHOULD include a Warning header field.

[RFC3261-14-34, 35]

A UAS rejecting an offer contained in an INVITE SHOULD return a 488 (Not Acceptable Here) response. Such a response SHOULD include a Warning header field value explaining why the offer was rejected.

#### 4.4.7 UA-5-2-5 - No ACK is received for re-INVITE (Caller)

##### [NAME]

UA-5-2-5 - No ACK is received for re-INVITE (Caller)

##### [PURPOSE]

Verify that a NUT (caller) properly processes when not receiving any ACK for a re-INVITE.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

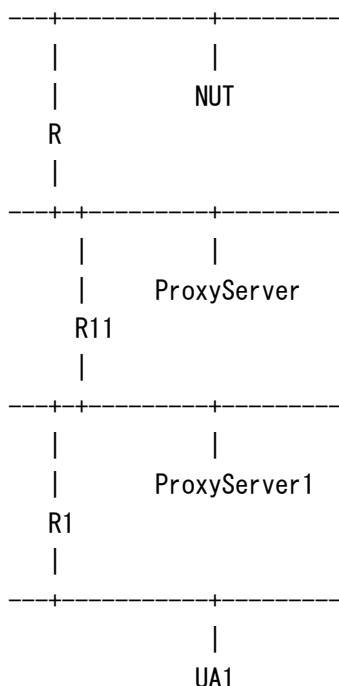
##### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

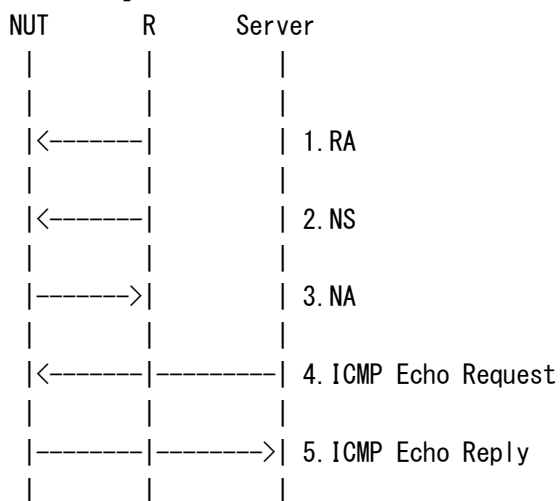
##### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

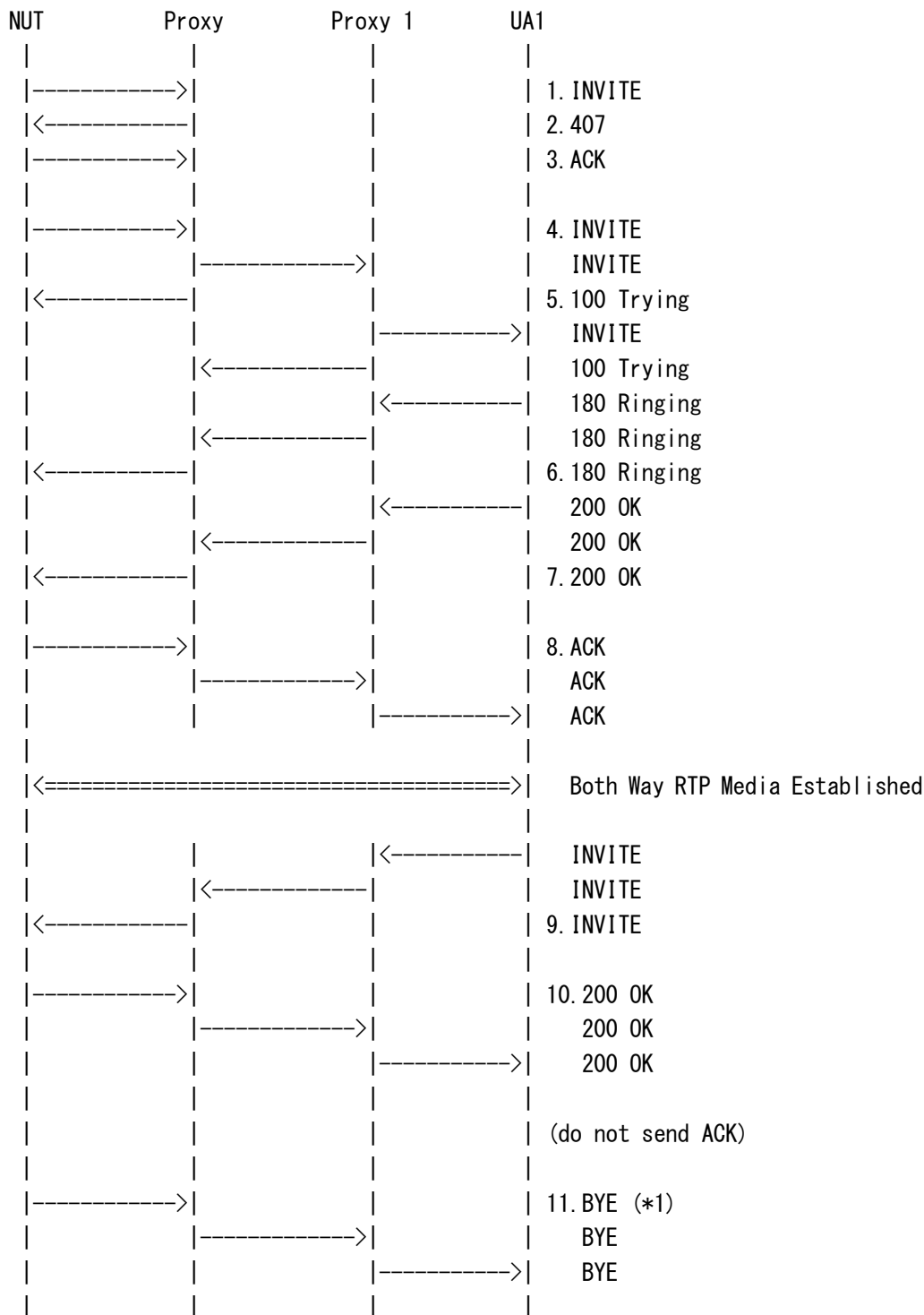


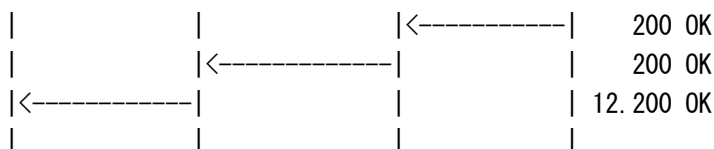
1. Send Router Advertisement.
2. Send Neighbor Solicitation.



3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Send INVITE.
10. Receive 200 OK.
11. Receive BYE. (\*1)
12. Send 200 OK.

#### [OBSERVABLE RESULTS]

\*1:BYE request from NUT.

Should send BYE if there is no ACK response for 2xx request. [RFC3261-14-16]

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_BYE  
Request-URI: Must be the URI of Contact in "7.200" response.  
[RFC3261-12-47]

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_BYE

\* To  
tag-param: Must equal that contained in the To header field of "7.200"  
response. [RFC3261-12-35]



**\* From**

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-37]

**\* Route**

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters.[RFC3261-12-48]

**- Bodies:**

See generic\_BYE

**[REFERENCE]**

[RFC3261-14-16]

14.2 UAS Behavior

If a UAS generates a 2xx response and never receives an ACK, it SHOULD generate a BYE to terminate the dialog.

**4.4.8 UA-5-2-6 - No ACK is received for re-INVITE (Callee)**

**[NAME]**

UA-5-2-6 - No ACK is received for re-INVITE (Callee)

**[PURPOSE]**

Verify that a NUT (callee) properly processes when not receiving any ACK for a re-INVITE.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

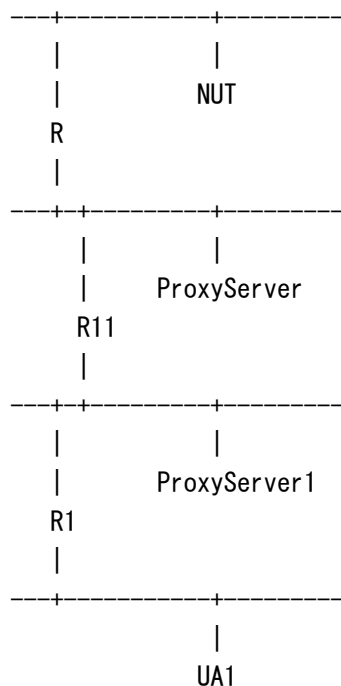
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

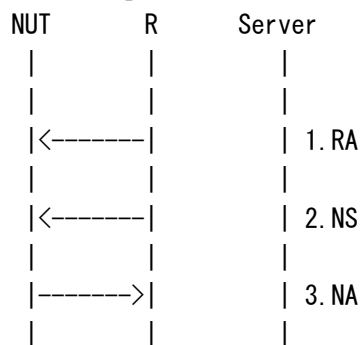
#### [TOPOLOGY]

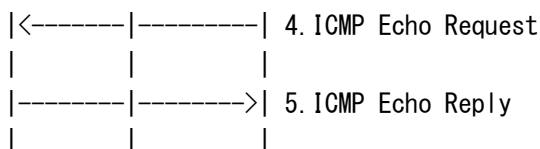


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

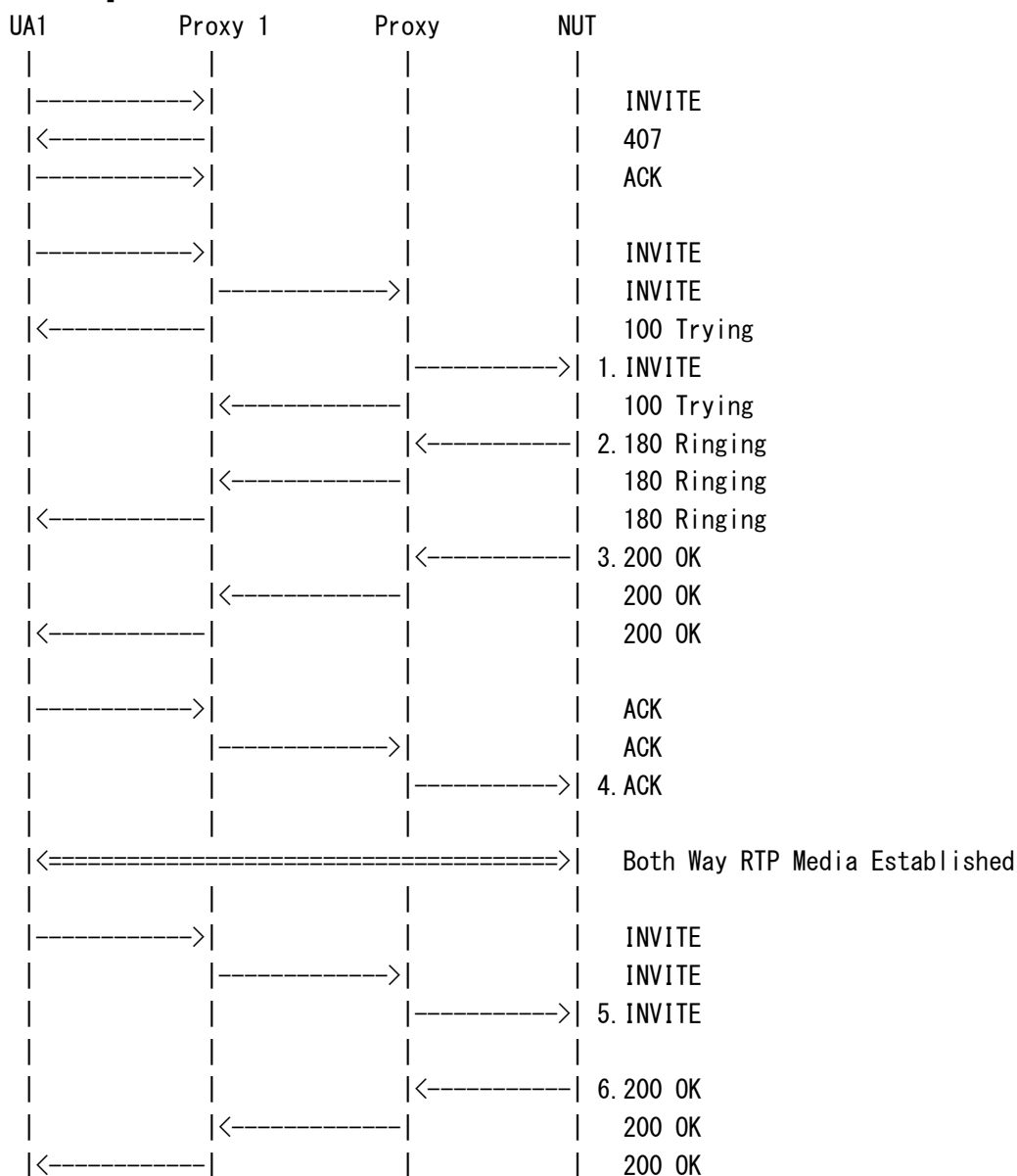
#### [INITIALIZATION]

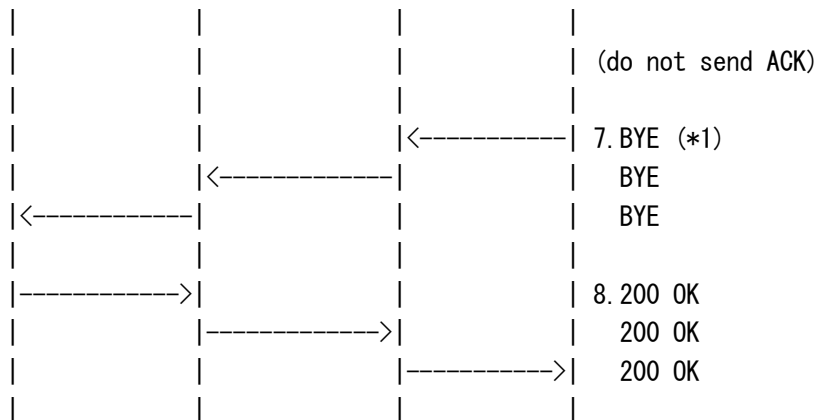




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





- [OBSERVABLE RESULTS]**

Should send BYE if there is no ACK response for 2xx request. [RFC3261-14-16]

As a SIP request,

See `generic_request`

See generic\_BYE

- Header fields:

See `generic_request`

- inside of a dialog

See generic\_BYE

271



tag-param: Must equal that contained in the To header field of "7.200" response. [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters.[RFC3261-12-48]

- Bodies:

See generic\_BYE

**[REFERENCE]**

[RFC3261-14-16]

14.2 UAS Behavior

If a UAS generates a 2xx response and never receives an ACK, it SHOULD generate a BYE to terminate the dialog.

**4.4.9 UA-5-2-7 - re-INVITE without offer (Caller)**

**[NAME]**

UA-5-2-7 - re-INVITE without offer (Caller)

**[PURPOSE]**

Verify that a NUT (caller) properly processes a re-INVITE without an offer.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

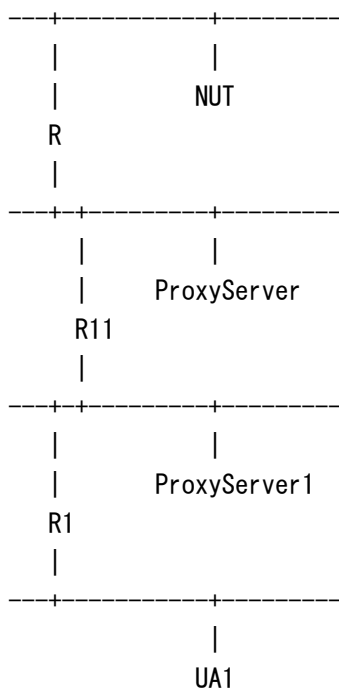
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

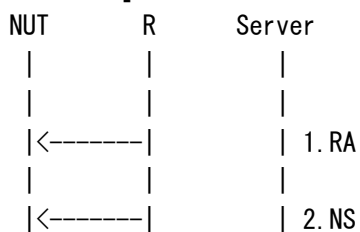
# [TOPOLOGY]



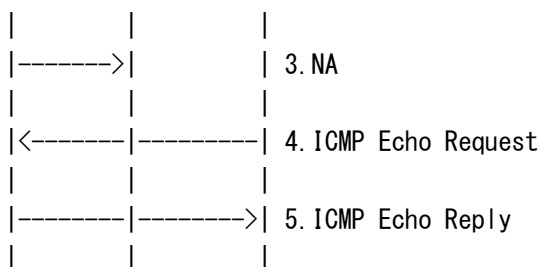
# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

# [INITIALIZATION]

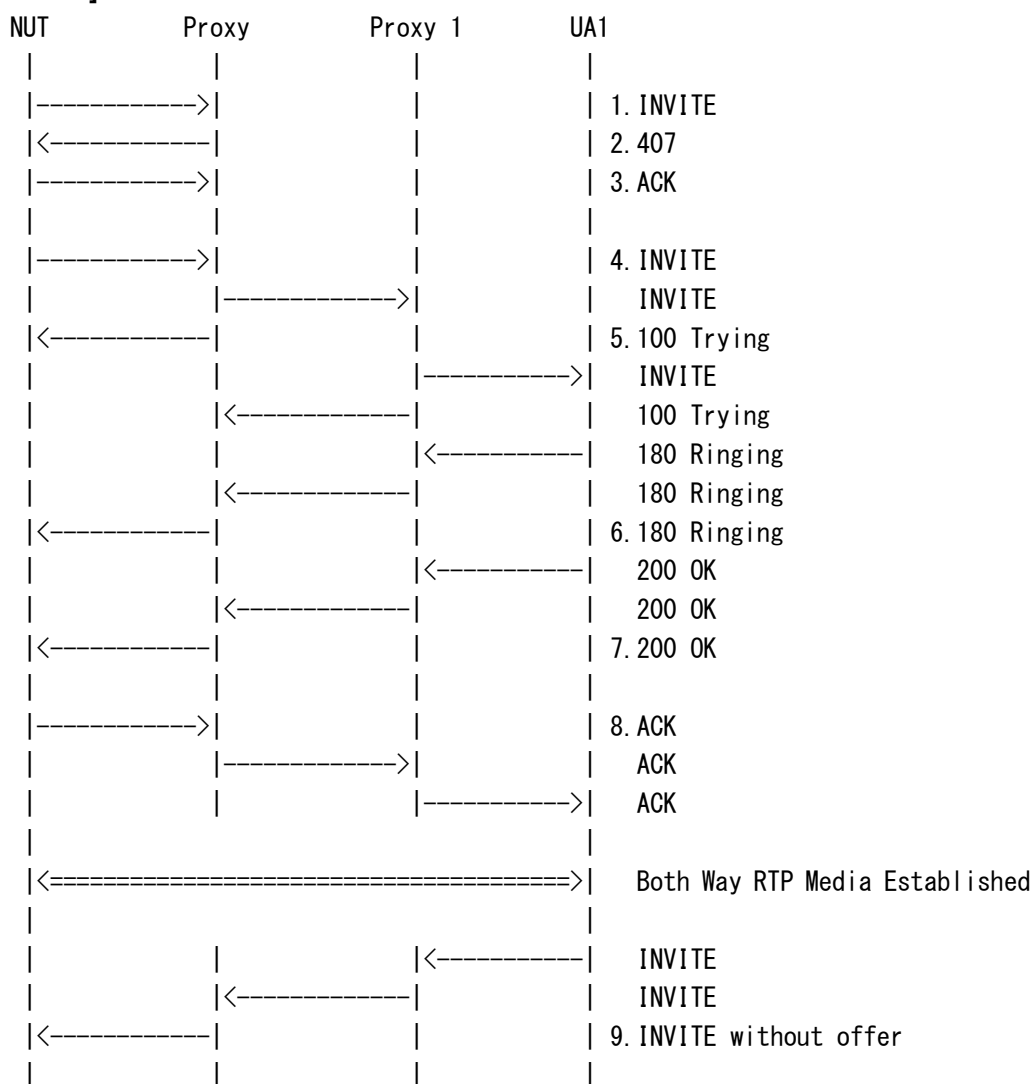


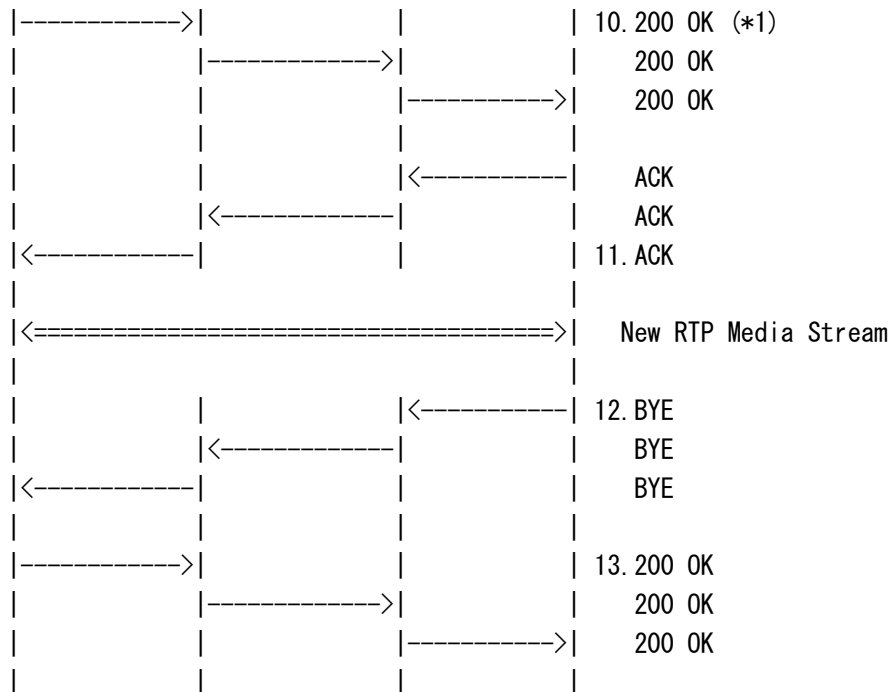




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Send INVITE.
10. Receive 200 OK. (\*1)
11. Send ACK.
12. Send BYE.
13. Receive 200 OK.

#### [OBSERVABLE RESULTS]

\*1:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response



See generic\_200-for-INVITE

Status-Code: Must be "200". [RFC3261 12.1,13, 21.2]

- Header fields:

See generic\_response

- inside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

"m=" line: Must include at least the same "m=" line as offer in "4.INVITE". [RFC3261-14-19] [RFC3264-6-2,3,4] [RFC4566 5.14]

"a=" line: Must include at least the same "a=" line as offer in "4.INVITE". [RFC3261-14-19]

"o=" line:

<version>: Must be incremented by one from <version> of previous SDP, Unless the same as offer in "4.INVITE". [RFC3261-14-19]

## [REFERENCE]

[RFC3261-14-17,18,19,20]

### 14.2 UAS Behavior

A UAS providing an offer in a 2xx (because the INVITE did not contain an offer) SHOULD construct the offer as if the UAS were making a brand new call, subject to the constraints of sending an offer that updates an existing session, as described in [13] in the case of SDP. Specifically, this means that it SHOULD include as many media formats and media types that the UA is willing to support. The UAS MUST



ensure that the session description overlaps with its previous session description in media formats, transports, or other parameters that require support from the peer. This is to avoid the need for the peer to reject the session description. If, however, it is unacceptable to the UAC, the UAC SHOULD generate an answer with a valid session description, and then send a BYE to terminate the session.

#### 4.4.10 UA-5-2-8 - re-INVITE without offer (Callee)

##### [NAME]

UA-5-2-8 - re-INVITE without offer (Callee)

##### [PURPOSE]

Verify that a NUT (callee) properly processes a re-INVITE without an offer.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

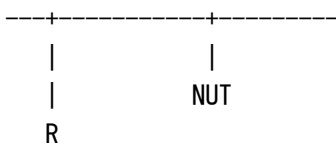
##### [PARAMETER]

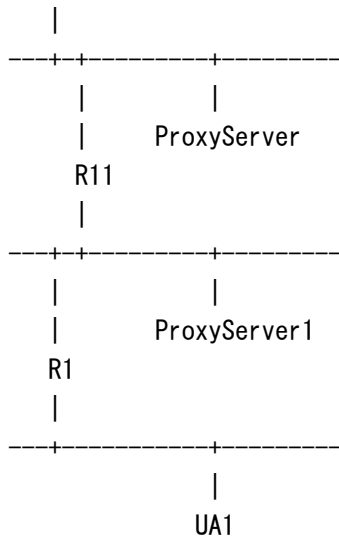
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

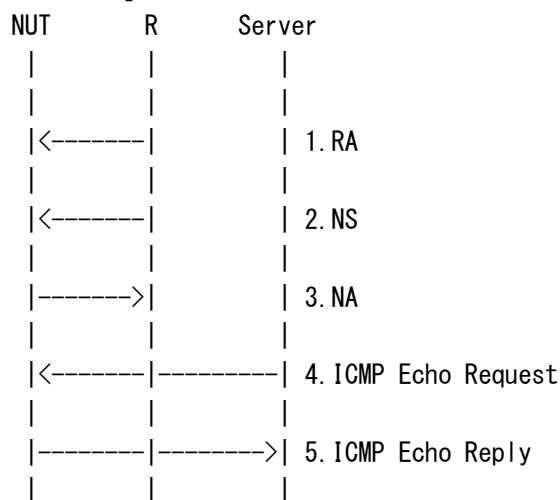




#### [CONFIGURATION for NUT]

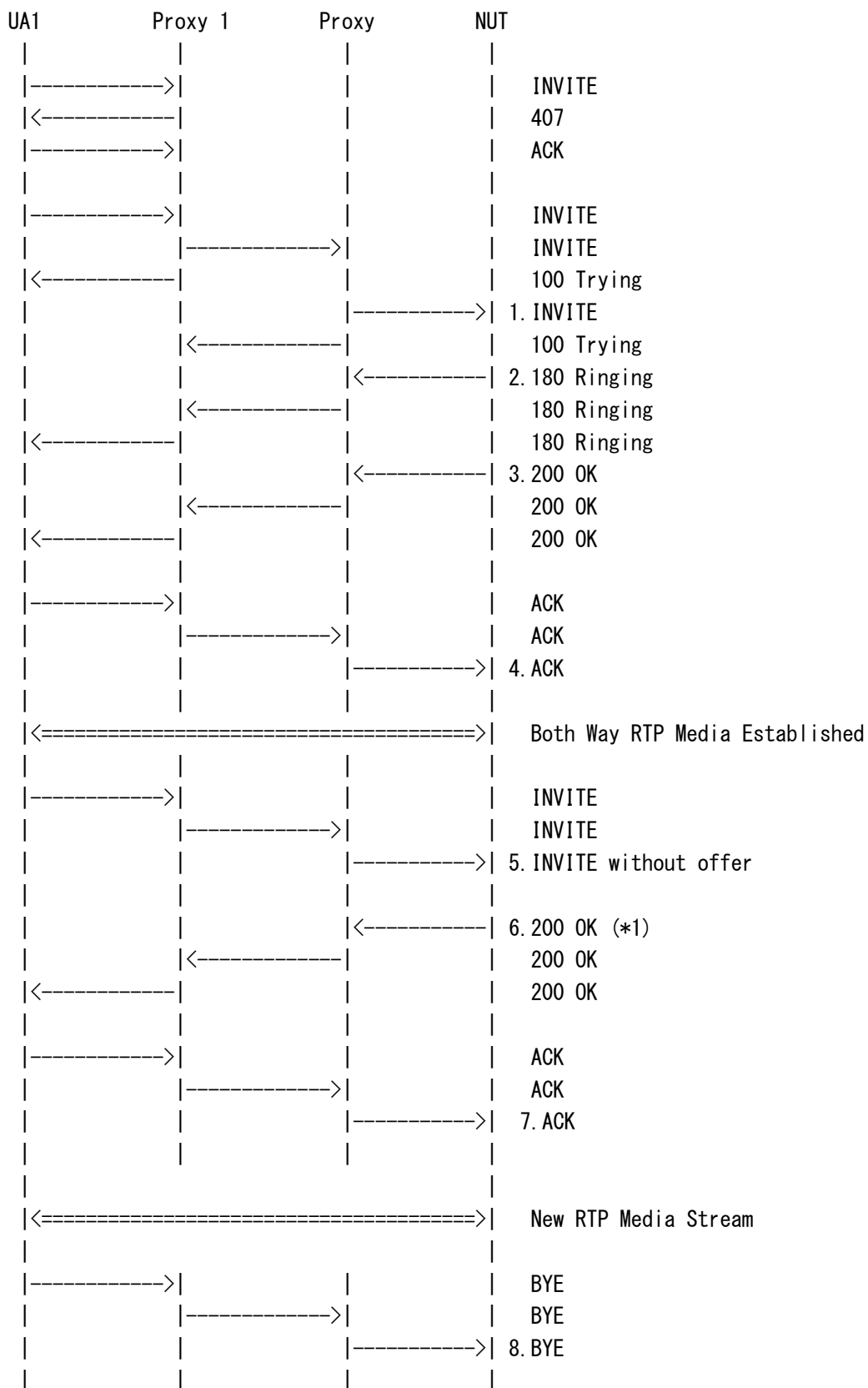
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

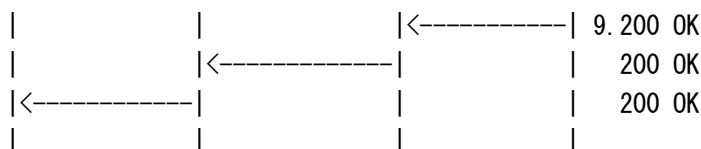
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send INVITE.
6. Receive 200 OK. (\*1)
7. Send ACK.
8. Send BYE.
9. Receive 200 OK.

#### [OBSERVABLE RESULTS]

\*1:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- inside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq-2]  
Must copy all Record-Route header field values from the request into the



response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

"m=" line: Must include at least the same "m=" line as offer in "3.200" response. [RFC3261-14-19] [RFC3264-6-2,3,4][RFC4566 5.14]

"a=" line: Must include at least the same "a=" line as offer in "3.200" response.  
[RFC3261-14-19]

"o=" line:

<version>: Must be incremented by one from <version> of previous SDP,  
Unless the same as offer in "3.200" response. [RFC3261-14-19]

## [REFERENCE]

[RFC3261-14-17, 18, 19, 20]

### 14.2 UAS Behavior

A UAS providing an offer in a 2xx (because the INVITE did not contain an offer) SHOULD construct the offer as if the UAS were making a brand new call, subject to the constraints of sending an offer that updates an existing session, as described in [13] in the case of SDP. Specifically, this means that it SHOULD include as many media formats and media types that the UA is willing to support. The UAS MUST ensure that the session description overlaps with its previous session description in media formats, transports, or other parameters that require support from the peer. This is to avoid the need for the peer to reject the session description. If, however, it is unacceptable to the UAC, the UAC SHOULD generate an answer with a valid session description, and then send a BYE to terminate the session.

## 4.4.11 UA-5-2-9 - Receipt of 491 response to re-INVITE (Caller)

### [NAME]

UA-5-2-9 - Receiving 491 response to re-INVITE (Caller)

### [PURPOSE]

Verify that a NUT (caller) properly processes when receiving a 491 response for a re-INVITE.

### [REQUIREMENT]



NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

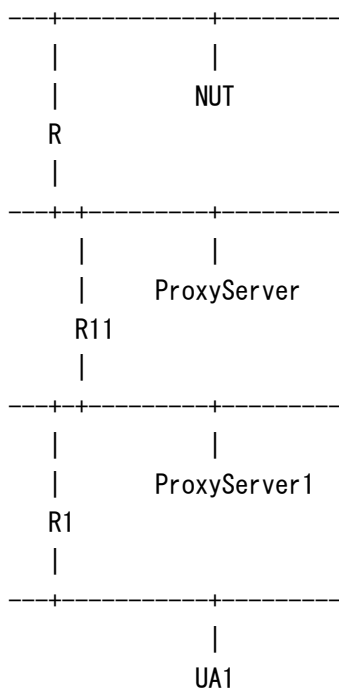
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

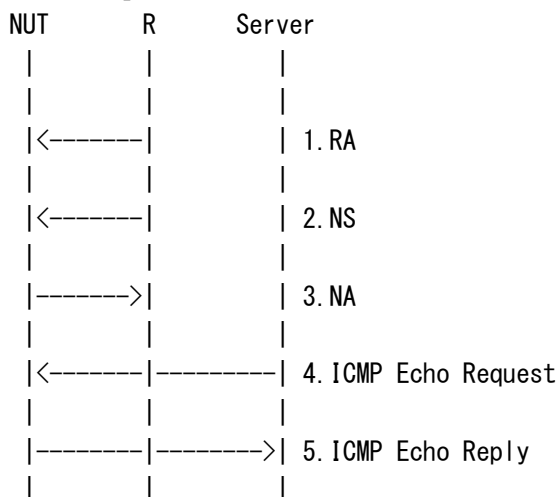


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

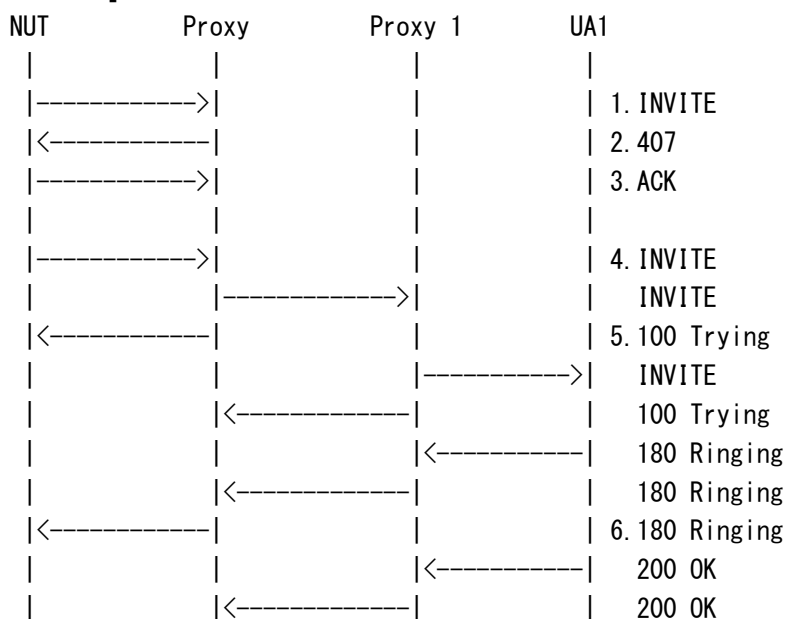
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

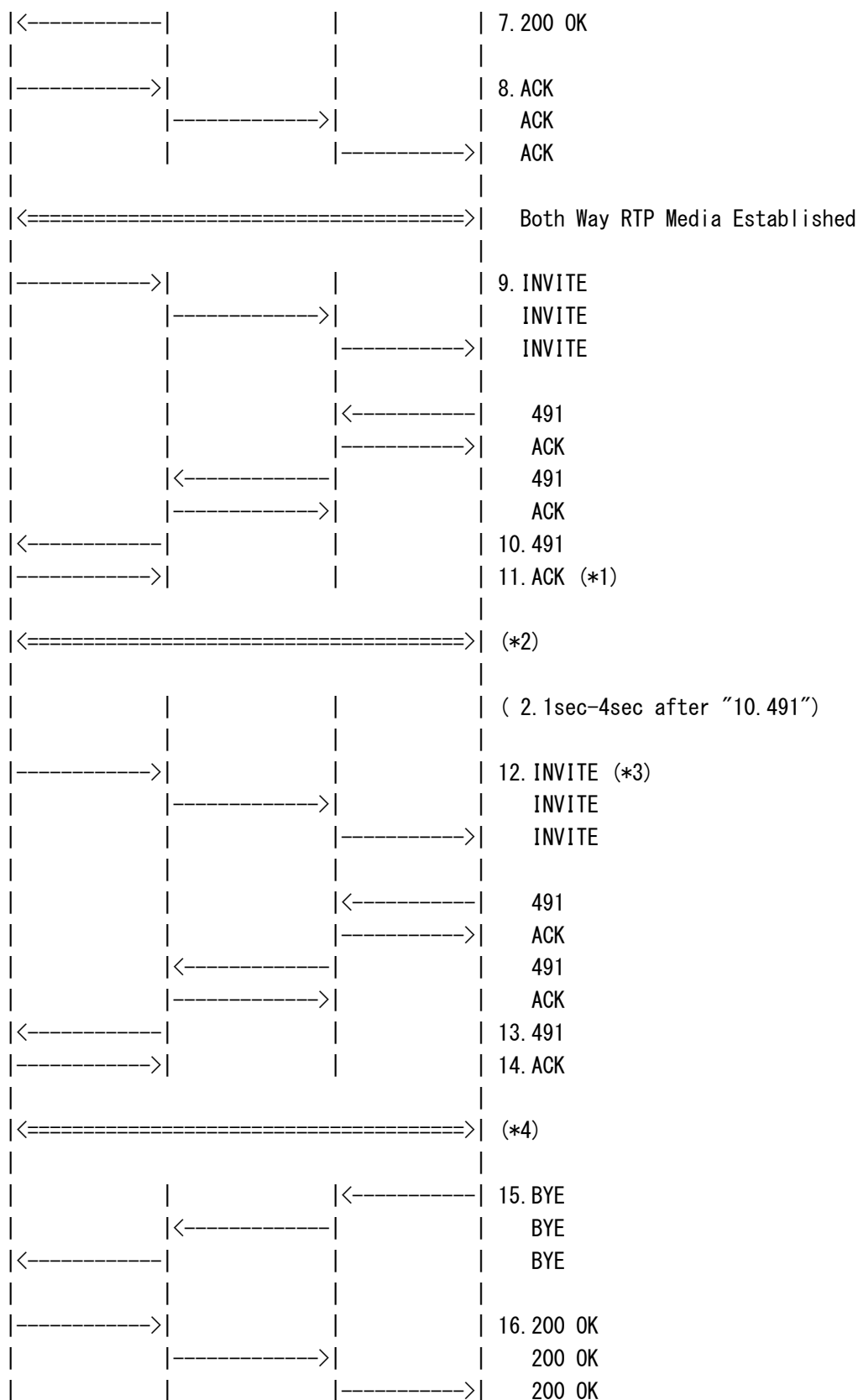
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





- |  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|
1. Receive INVITE.
  2. Send 407.
  3. Receive ACK.
  4. Receive INVITE.
  5. Send 100 Trying.
  6. Send 180 Ringing.
  7. Send 200 OK.
  8. Receive ACK.
  9. Receive INVITE.
  10. Send 491 Request Pending.
  11. Receive ACK. (\*1)
  - (\*2)
  12. Receive INVITE. (\*3)
  13. Send 491 Request Pending.
  14. Receive ACK.
  - (\*4)
  15. Send BYE.
  16. Receive 200 OK.

#### **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

\* Route  
Must include a Route header field. [RFC3261-12-48]  
route-param: Must contain the route set values(set by Record-Route



header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:
  - See generic\_ACK
  - See generic\_non2xx-ACK

\*2:after ACK request from NUT.

RTP session Must not be changed. [RFC3261-14-7]

\*3:INVITE request from NUT.

INVITE Should be sent after a randomly chosen time between 2.1 and 4 seconds in units of 10ms. [RFC3261-14-8, RFC3261-14-9]

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_re-INVITE

Request-URI: Must be the URI of Contact in "1.INVITE". [RFC3261-12-47]

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_re-INVITE

\* To  
tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-35]

\* From  
tag-param: Must equal that contained in the To header field of "2.200" response. [RFC3261-12-37]

\* Route



Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values (set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_request

See generic\_re-INVITE

"a=" line: must either include an "a=sendrecv" attribute, or omit it. [RFC3264 5.1][RFC4566 6]

"o=" line:

<sess-version>: Must be incremented by one from <sess-version> of previous SDP. [RFC3264-8-1, 2][RFC4566 5.2]

"m=" line: Must include the same m line as SDP in INVITE included.

(tester checks only the number of "m=" line.) [RFC3264-6-2,3,4][RFC4566 5.14]

\*4:after ACK request from NUT.

RTP session Must not be changed. [RFC3261-14-7]

## [REFERENCE]

[RFC3261-14-7, 8, 9]

### 14.1 UAC Behavior

If a UA receives a non-2xx final response to a re-INVITE, the session parameters MUST remain unchanged, as if no re-INVITE had been issued. Note that, as stated in Section 12.2.1.2, if the non-2xx final response is a 481 (Call/Transaction Does Not Exist), or a 408 (Request Timeout), or no response at all is received for the re-INVITE (that is, a timeout is returned by the INVITE client transaction), the UAC will terminate the dialog.

(snip)

If a UAC receives a 491 response to a re-INVITE, it SHOULD start a timer with a value T chosen as follows:

1. If the UAC is the owner of the Call-ID of the dialog ID (meaning it generated the value), T has a randomly chosen value between 2.1 and 4 seconds in units of 10 ms.



2. If the UAC is not the owner of the Call-ID of the dialog ID, T has a randomly chosen value of between 0 and 2 seconds in units of 10 ms.

When the timer fires, the UAC SHOULD attempt the re-INVITE once more, if it still desires for that session modification to take place. For example, if the call was already hung up with a BYE, the re-INVITE would not take place.

#### 4.4.12 UA-5-2-10 - Receipt of 491 response for re-INVITE (Callee)

##### [NAME]

UA-5-2-10 – Receipt of 491 response for re-INVITE (Callee)

##### [PURPOSE]

Verify that a NUT (callee) properly processes when receiving a 491 response for a re-INVITE.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

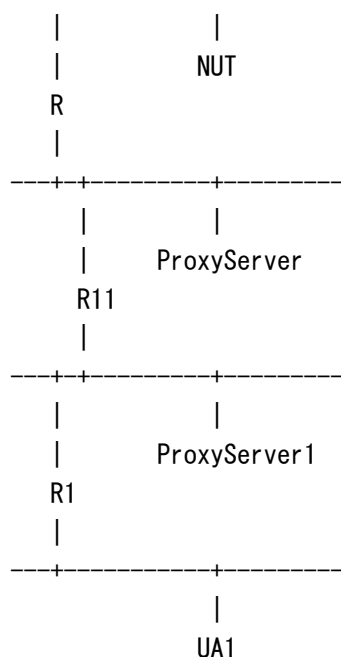
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

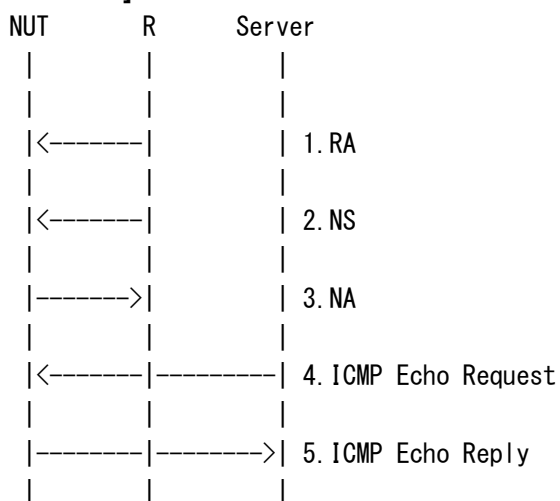
-----+-----+-----



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

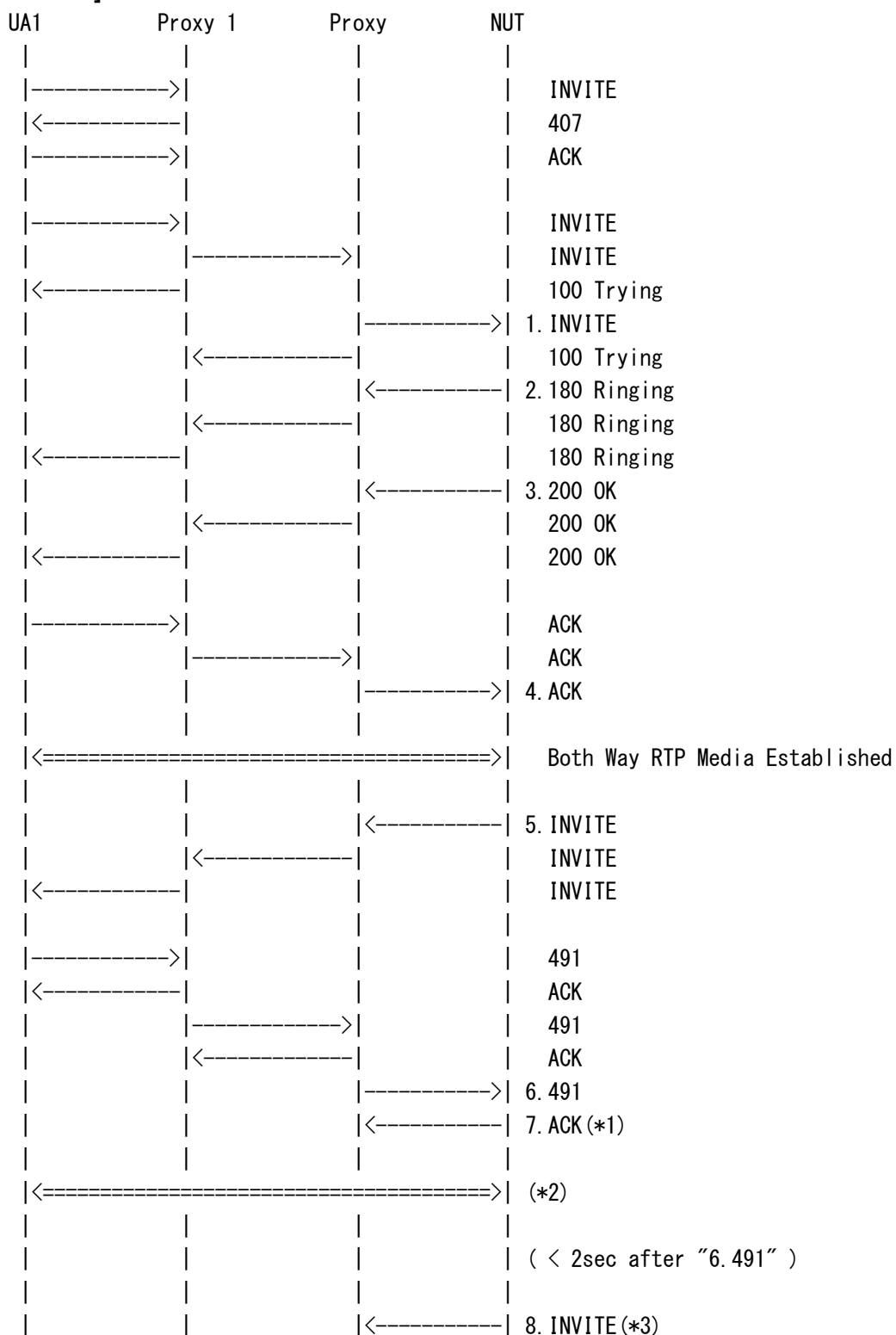


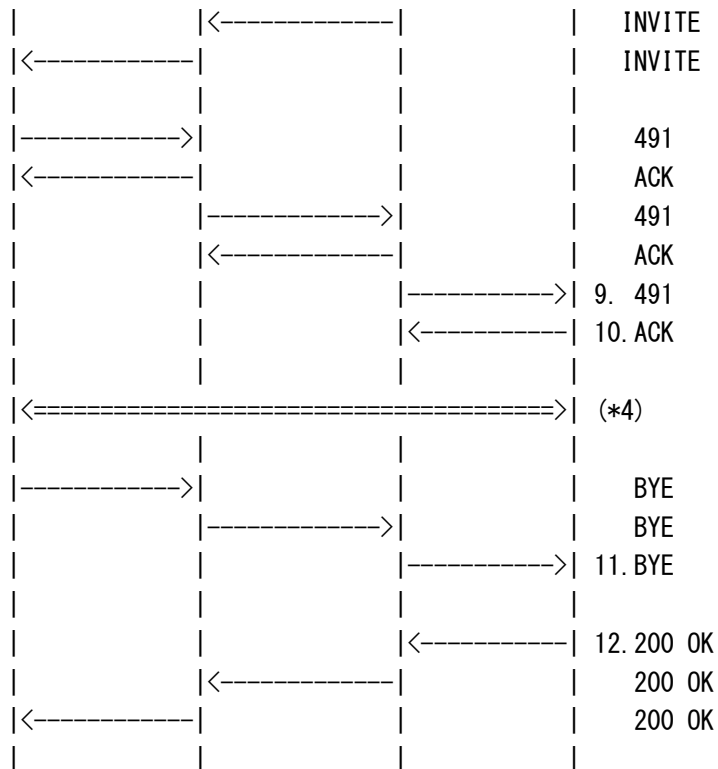
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.



## 5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Receive INVITE.
6. Send 491 Request Pending.
7. Receive ACK. (\*1)
- (\*2)
8. Receive INVITE. (\*3)
9. Send 491 Request Pending.
10. Receive ACK.
- (\*4)
11. Send BYE.
12. Receive 200 OK.

#### [OBSERVABLE RESULTS]

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:

See generic\_request

See generic\_ACK

See generic\_non2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_non2xx-ACK

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values(set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_ACK

See generic\_non2xx-ACK

\*2:after ACK request from NUT.

RTP session Must not be changed. [RFC3261-14-7]

\*3:INVITE request from NUT.

INVITE Should be sent after a randomly chosen time between 0 and 2 seconds in units of 10ms. [RFC3261-14-8, RFC3261-14-9]

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_re-INVITE

Request-URI: Must be the URI of Contact in "1.INVITE". [RFC3261-12-47]



- Header fields:

See generic\_request

- inside of a dialog

See generic\_re-INVITE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "2.200" response. [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values (set by Record-Route header field in 1st-INVITE) in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_request

See generic\_re-INVITE

"a=" line: must either include an "a=sendrecv" attribute, or omit it. [RFC3264 5.1][RFC4566 6]

"o=" line:

<sess-version>: Must be incremented by one from <sess-version> of previous SDP. [RFC3264-8-1, 2][RFC4566 5.2]

"m=" line: Must include the same m line as SDP in INVITE included.

(tester checks only the number of "m=" line.) [RFC3264-6-2,3,4][RFC4566 5.14]

\*4:after ACK request from NUT.

RTP session Must not be changed. [RFC3261-14-7]

## [REFERENCE]

[RFC3261-14-7, 8, 9]

14.1 UAC Behavior



If a UA receives a non-2xx final response to a re-INVITE, the session parameters **MUST** remain unchanged, as if no re-INVITE had been issued. Note that, as stated in Section 12.2.1.2, if the non-2xx final response is a 481 (Call/Transaction Does Not Exist), or a 408 (Request Timeout), or no response at all is received for the re-INVITE (that is, a timeout is returned by the INVITE client transaction), the UAC will terminate the dialog.

(snip)

If a UAC receives a 491 response to a re-INVITE, it **SHOULD** start a timer with a value T chosen as follows:

1. If the UAC is the owner of the Call-ID of the dialog ID (meaning it generated the value), T has a randomly chosen value between 2.1 and 4 seconds in units of 10 ms.
2. If the UAC is not the owner of the Call-ID of the dialog ID, T has a randomly chosen value of between 0 and 2 seconds in units of 10 ms.

When the timer fires, the UAC **SHOULD** attempt the re-INVITE once more, if it still desires for that session modification to take place. For example, if the call was already hung up with a BYE, the re-INVITE would not take place.

## 4.5 Authentication

### 4.5.1 UA-6-1-5 - BYE request with user authentication

#### **[NAME]**

UA-6-1-5 - BYE request with user authentication

#### **[PURPOSE]**

Verify that a NUT properly processes a BYE request with user authentication.

#### **[REQUIREMENT]**

Only when a NUT supports user authentication for BYE.

#### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

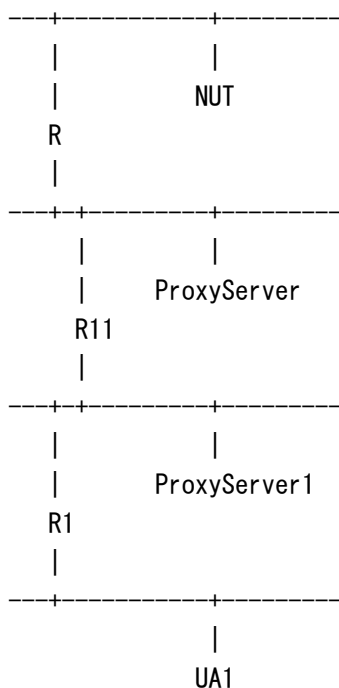
#### **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

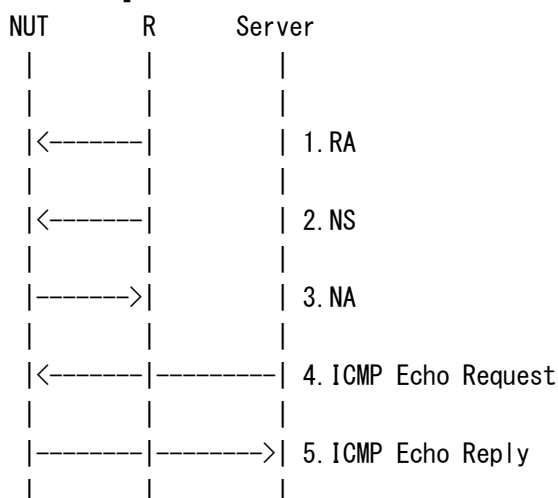
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

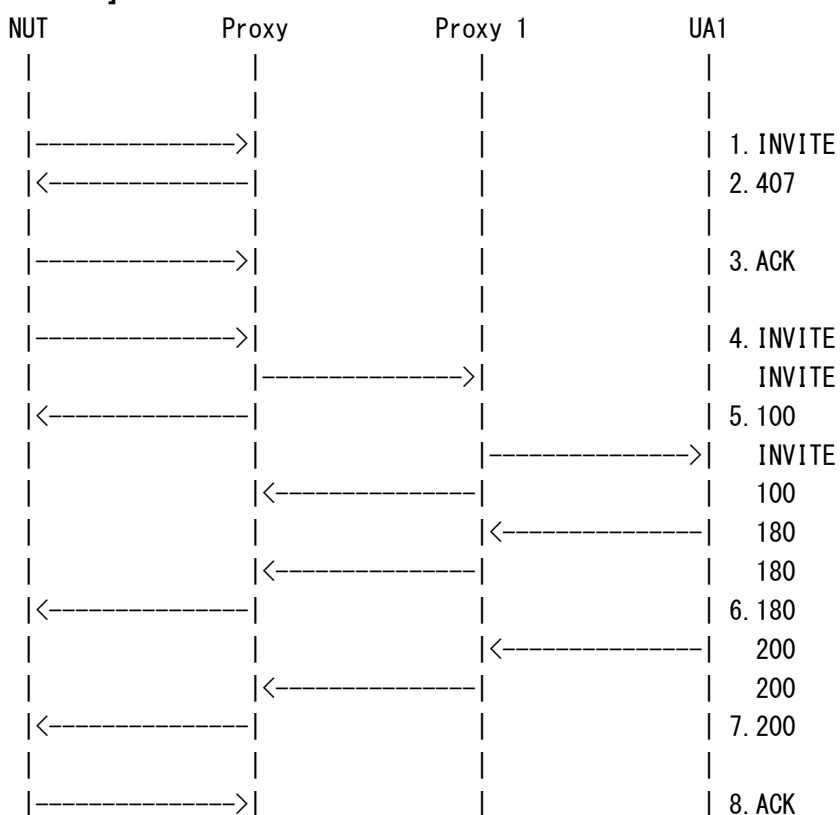
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

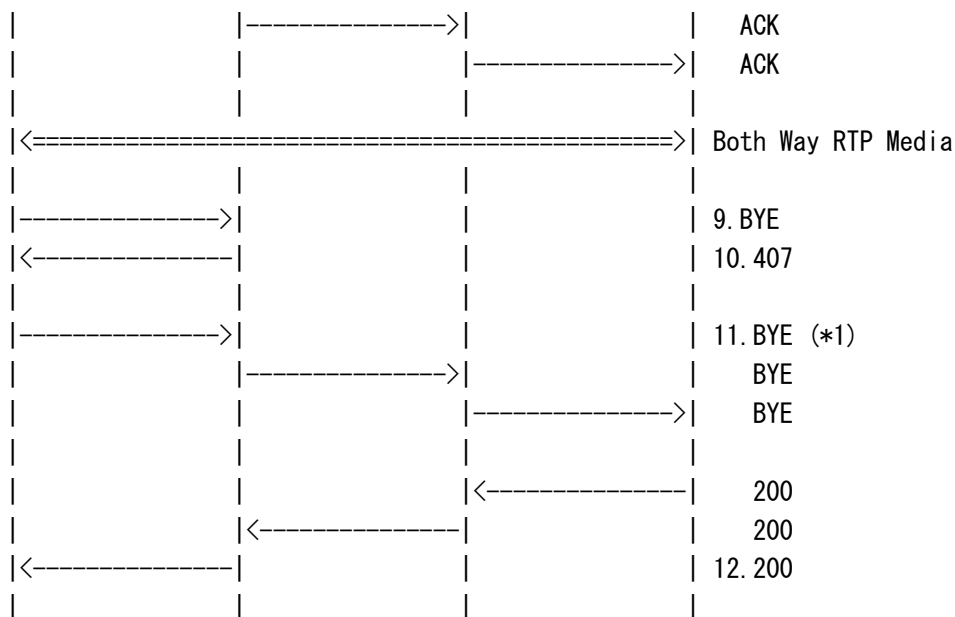
## [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

## [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Receive BYE.
10. Send 407 Proxy Authorization Required.
11. Receive BYE(with Proxy-Authorization header field). (\*1)
12. Send 200 OK.

\* 9-10 messages are optional, so you can send 11.BYE directly.

#### [OBSERVABLE RESULTS]

\*1:BYE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_BYE  
Request-URI: Must be the URI of Contact in 200 response. [RFC3261-12-47]





- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the To header field of "7.200" response.  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

See generic\_digest-auth

- Bodies:

See generic\_BYE

#### **[REFERENCE]**

NONE

### **4.5.2 UA-6-1-6 - re-INVITE with user authentication**

#### **[NAME]**

UA-6-1-6 - re-INVITE with user authentication

#### **[PURPOSE]**

Verify that a NUT properly processes a re-INVITE request with user authentication.

#### **[REQUIREMENT]**

Only when a NUT supports user authentication for re-INVITE.

Only when a NUT supports hold using re-INVITE.

#### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

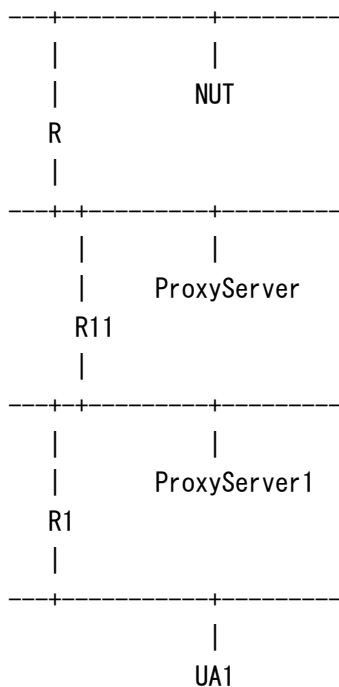
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

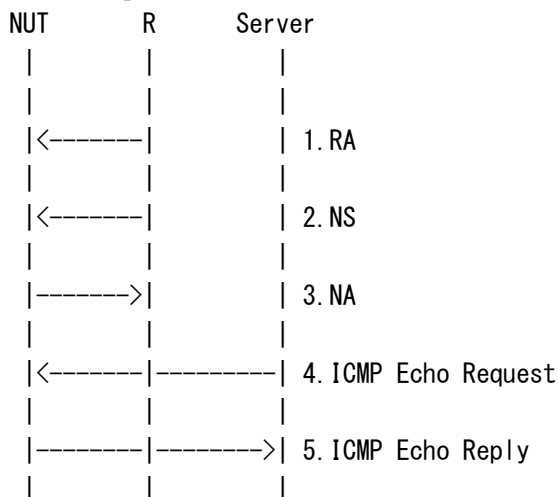


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr

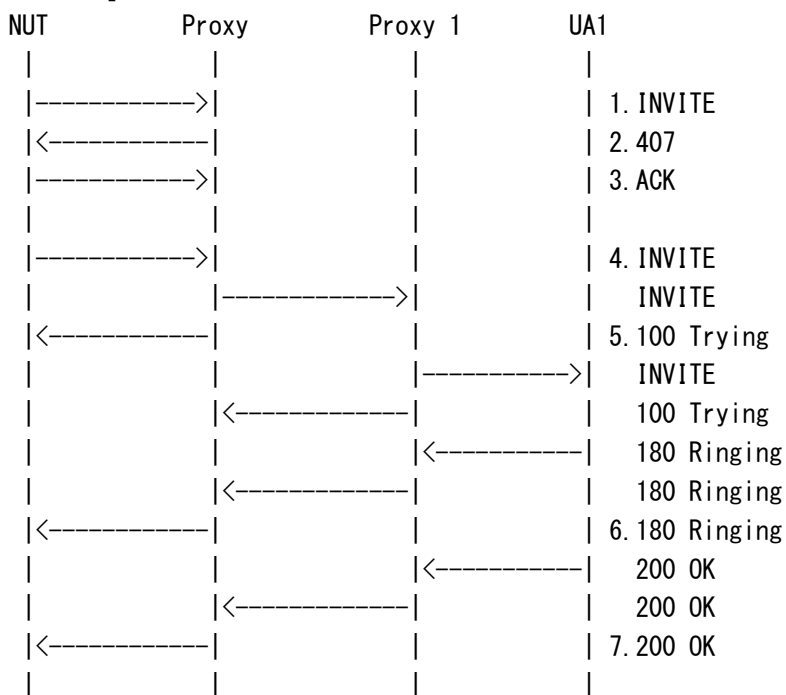
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

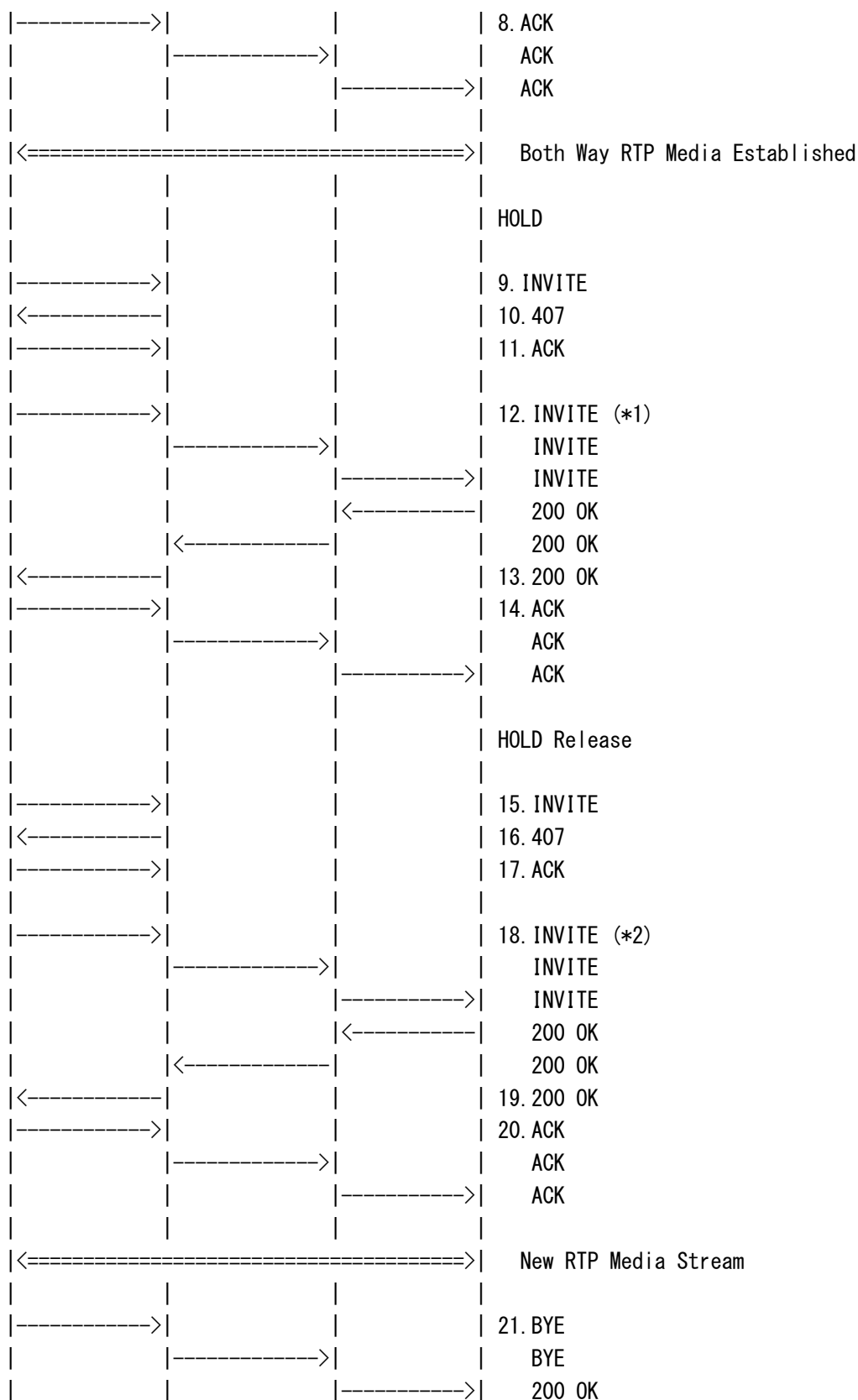
### [INITIALIZATION]

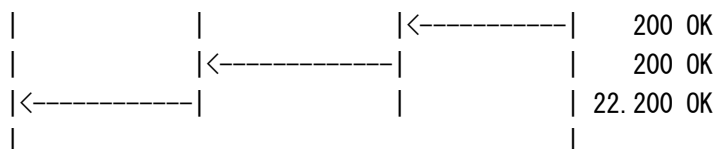


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]







1. Receive INVITE.
2. Send 407.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Receive INVITE.
10. Send 407.
11. Receive ACK.
12. Receive INVITE. (\*1)
13. Send 200 OK.
14. Receive ACK.
15. Receive INVITE.
16. Send 407.
17. Receive ACK.
18. Receive INVITE. (\*2)
19. Send 200 OK.
20. Receive ACK.
21. Receive BYE.
22. Send 200 OK.

### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_re-INVITE

- Header fields:  
See generic\_request

- inside of a dialog



See generic\_re-INVITE

\* To

tag-param: Must equal that contained in the From header field of "7.200 OK" response. [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "1.INVITE". [RFC3261-12-37]

\* Route

MUST include a Route header field. [RFC3261-12-48]

route-param: MUST contain the route set values in order, including all parameters.[RFC3261-12-48]

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

See generic\_digest-auth

- Bodies:

See generic\_re-INVITE

"a=" line: "a=sendonly" must be appended into SDP.

"o=" line:

<version>: Must be incremented by one from <version> of previous SDP.

"m=" line: Must include the same m line as SDP in INVITE included.

(tester checks only the number of "m=" line.) [RFC3264-6-2,3,4][RFC4566 5.14]

\*2:INVITE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_re-INVITE

- Header fields:

See generic\_request

- inside of a dialog

See generic\_re-INVITE



- \* To
  - tag-param: Must equal that contained in the From header field of "7.200 OK" response. [RFC3261-12-35]
- \* From
  - tag-param: Must equal that contained in the To header field of "1.INVITE". [RFC3261-12-37]
- \* Route
  - MUST include a Route header field. [RFC3261-12-48]
  - route-param: MUST contain the route set values in order, including all parameters.[RFC3261-12-48]
- \* Proxy-Authorization
  - Must exist. [RFC3261.22.3],[RFC3261-22-22]
  - See generic\_digest-auth
- Bodies:
  - See generic\_re-INVITE
  - "a=" line: must either include an "a=sendrecv" attribute, or omit it. [RFC3264 5.1][RFC4566 6]
  - "o=" line:
    - <sess-version>: Must be incremented by one from <sess-version> of previous SDP. [RFC3264-8-1, 2][RFC4566 5.2]
  - "m=" line: Must include the same m line as SDP in INVITE included.  
(tester checks only the number of "m=" line.) [RFC3264-6-2,3,4][RFC4566 5.14]

#### **[REFERENCE]**

NONE

### **4.5.3 UA-6-1-7 - REGISTER with Digest Authentication without qop**

#### **[NAME]**

UA-6-1-7 - REGISTER with Digest Authentication without qop

#### **[PURPOSE]**

Verify that a NUT properly processes a REGISTER request with digest authentication without qop parameter.

#### **[REQUIREMENT]**

NONE

**[TARGET]**

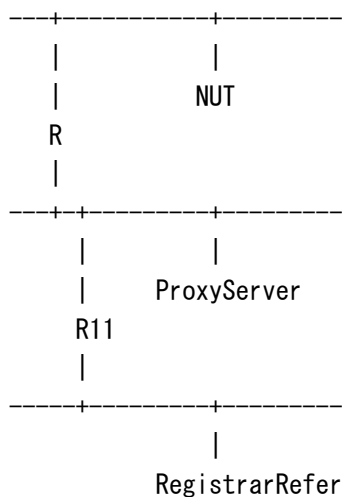
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

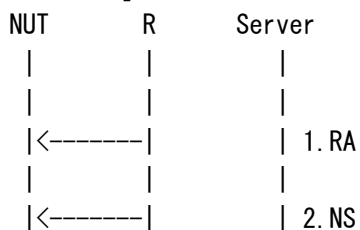
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com

**[ADDRESS]**

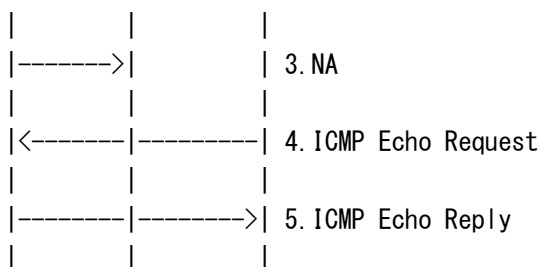
NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
Registrar(IPv6)	3ffe:501:ffff:50::60/64

**[TOPOLOGY]****[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Registrar	sip:reg.under.test.com
Registrar	3ffe:501:ffff:50::60/64 (IPv6)

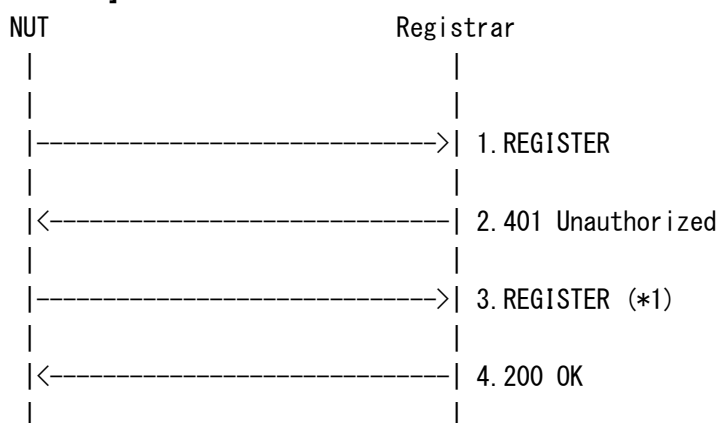
**[INITIALIZATION]**





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive REGISTER.
2. Send 401 Unauthorized (without qop parameter in WWW-Authenticate).
3. Receive REGISTER (without qop parameter in Authorization). (\*1)
4. Send 200 OK.

#### 2. 401 Unauthorized Registrar -> NUT

(snip)

WWW-Authenticate: Digest realm="reg.under.test.com",  
 nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
 stale=FALSE, algorithm=MD5

(snip)

\*without qop parameter.



### 3. REGISTER NUT -> Registrar

(snip)

Authorization: Digest username="NUT", realm="reg.under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:reg.under.test.com",  
response="c88605a792c947d7a00fe44119ac7373"

(snip)

#### [OBSERVABLE RESULTS]

\*1: REGISTER request from NUT

As a SIP Message,

See generic\_request

As a SIP request,

- Request-Line:

See generic\_request

See generic\_REGISTER

- Header fields:

See generic\_request

- outside of a dialog

See generic\_REGISTER

\* Call-ID

callid: Must be the same as that of \*1 message. [RFC3261-8-11] [RFC3261-10-8, 17]

\* Cseq

1\*DIGIT: Must be incremented from \*1 message by one with the same Call-ID.  
[RFC3261-10-9]

\* Contact

Must exist. [RFC3261.10.2.1]

contact-param: Must be the specified parameter. [RFC3261.10.2.1]

contact-param: "\*" MUST NOT be used unless the Expires header field is present with a value of "0". [RFC3261-10-15]

c-p-expires:

delta-seconds: Must not be "0". [RFC3261.10.2.2]

\* Expires

delta-seconds: Must not be "0" (when the expires parameter of a Contact header



field does not exist). [RFC3261.10.2.2]

\* Authorization

Must exist. [RFC3261 22.2]

See generic\_digest-auth

See generic\_digest-noqop with care

- Bodies:

See generic\_REGISTER

**[REFERENCE]**

[RFC3261-22-32, 35, 36, 37, 38]

22.4 The Digest Authentication Scheme [Page 200]

Since RFC 2543 is based on HTTP Digest as defined in RFC 2069 [39], SIP servers supporting RFC 2617 MUST ensure they are backwards compatible with RFC 2069. Procedures for this backwards compatibility are specified in RFC 2617. Note, however, that SIP servers MUST NOT accept or request Basic authentication.

8. RFC 2617 notes that a cnonce value MUST NOT be sent in an Authorization (and by extension Proxy-Authorization) header field if no qop directive has been sent. Therefore, any algorithms that have a dependency on the cnonce (including "MD5-Sess") require that the qop directive be sent. Use of the "qop" parameter is optional in RFC 2617 for the purposes of backwards compatibility with RFC 2069; since RFC 2543 was based on RFC 2069, the "qop" parameter must unfortunately remain optional for clients and servers to receive. However, servers MUST always send a "qop" parameter in WWW-Authenticate and Proxy-Authenticate header field values. If a client receives a "qop" parameter in a challenge header field, it MUST send the "qop" parameter in any resulting authorization header field.

RFC 2543 did not allow usage of the Authentication-Info header field (it effectively used RFC 2069). However, we now allow usage of this header field, since it provides integrity checks over the bodies and provides mutual authentication. RFC 2617 [17] defines mechanisms for backwards compatibility using the qop attribute in the request. These mechanisms MUST be used by a server to determine if the client supports the new mechanisms in RFC 2617 that were not specified in



RFC 2069.

[RFC2617-3-6, 7, 8, 9]

### 3.2.2 The Authorization Request Header field [Page 12]

cnonce

This MUST be specified if a qop directive is sent (see above), and MUST NOT be specified if the server did not send a qop directive in the WWW-Authenticate header field.

nonce-count

This MUST be specified if a qop directive is sent (see above), and MUST NOT be specified if the server did not send a qop directive in the WWW-Authenticate header field.

## 4.5.4 UA-6-1-8 - INVITE with Digest Authentication without qop

### [NAME]

UA-6-1-8 - INVITE with Digest Authentication without qop

### [PURPOSE]

Verify that a NUT properly processes an INVITE request with digest authentication without qop parameter.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

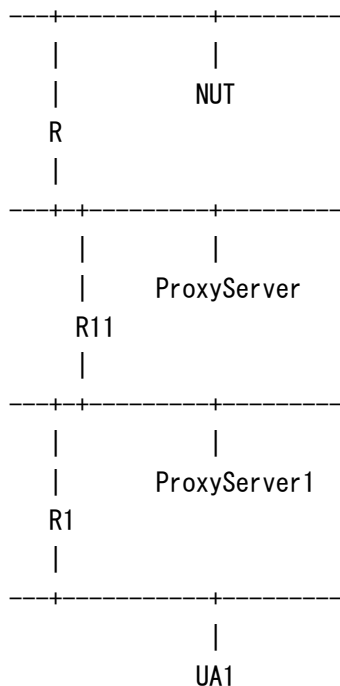
### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64



ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

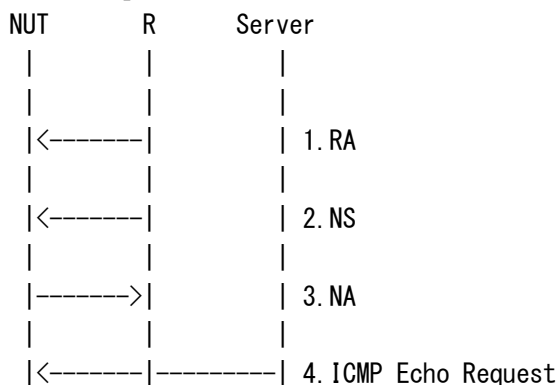
#### [TOPOLOGY]

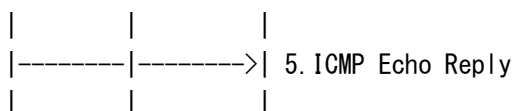


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

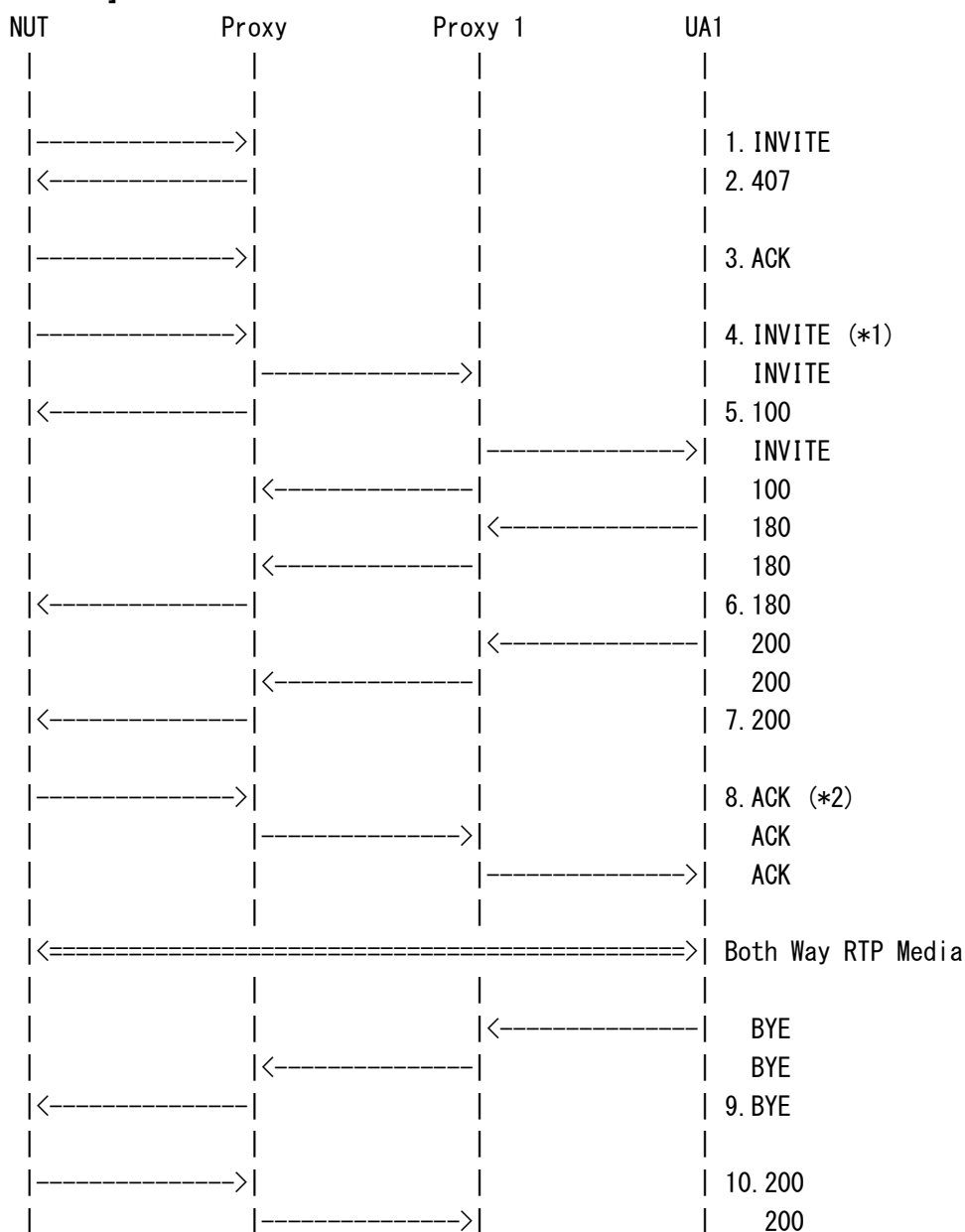
#### [INITIALIZATION]





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





| | |----->| 200  
| | |

1. Receive INVITE.
2. Send 407 Proxy Authorization Required  
(without qop parameter in Proxy-Authenticate).
3. Receive ACK.
4. Receive INVITE (without qop parameter in Proxy-Authorization). (\*1)
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK. (\*2)
9. Send BYE.
10. Receive 200 OK.

## 2. 407 Proxy Authorization Required Proxy -> NUT

(snip)

Proxy-Authenticate: Digest realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
stale=FALSE, algorithm=MD5

(snip)

\*without qop parameter.

## 4. INVITE NUT -> Proxy

(snip)

Proxy-Authorization: Digest username="NUT", realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:UA1@atlanta.example.com",  
response="af8098bf32569338c4eadbf7cc3fa3be"

(snip)

## 8. ACK NUT -> Proxy

(snip)

Proxy-Authorization: Digest username="NUT", realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:UA1@atlanta.example.com",  
response="af8098bf32569338c4eadbf7cc3fa3be"

(snip)

## [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

\* Call-ID  
callid: Should be the same as that of "1.INVITE" request, with case-sensitivity  
on. [RFC3261-8-62]

\* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
See generic\_digest-noqop

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*2:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK





- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

- \* Proxy-Authorization  
Must exist. [RFC3261-22.3],[RFC3261-22-22]  
Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]  
See generic\_digest-noqop with care

- \* Route  
Must exist. [ORq-2]  
Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:  
See generic\_ACK  
See generic\_2xx-ACK

## [REFERENCE]

[RFC3261-22-32, 35, 36, 37, 38]

22.4 The Digest Authentication Scheme [Page 200]

Since RFC 2543 is based on HTTP Digest as defined in RFC 2069 [39], SIP servers supporting RFC 2617 MUST ensure they are backwards compatible with RFC 2069. Procedures for this backwards compatibility are specified in RFC 2617. Note, however, that SIP servers MUST NOT accept or request Basic authentication.

8. RFC 2617 notes that a cnonce value MUST NOT be sent in an Authorization (and by extension Proxy-Authorization) header field if no qop directive has been sent. Therefore, any algorithms that have a dependency on the cnonce (including "MD5-Sess") require that the qop directive be sent. Use of the "qop" parameter is optional in RFC 2617 for the purposes of backwards compatibility with RFC 2069; since RFC 2543 was based on RFC 2069, the "qop" parameter must unfortunately remain optional for clients and servers to receive. However,



servers **MUST** always send a "qop" parameter in WWW-Authenticate and Proxy-Authenticate header field values. If a client receives a "qop" parameter in a challenge header field, it **MUST** send the "qop" parameter in any resulting authorization header field.

RFC 2543 did not allow usage of the Authentication-Info header field (it effectively used RFC 2069). However, we now allow usage of this header field, since it provides integrity checks over the bodies and provides mutual authentication. RFC 2617 [17] defines mechanisms for backwards compatibility using the qop attribute in the request. These mechanisms **MUST** be used by a server to determine if the client supports the new mechanisms in RFC 2617 that were not specified in RFC 2069.

[RFC2617-3-6, 7, 8, 9]

### 3.2.2 The Authorization Request Header field [Page 12]

cnonce

This **MUST** be specified if a qop directive is sent (see above), and **MUST NOT** be specified if the server did not send a qop directive in the WWW-Authenticate header field.

nonce-count

This **MUST** be specified if a qop directive is sent (see above), and **MUST NOT** be specified if the server did not send a qop directive in the WWW-Authenticate header field.

## 4.5.5 UA-6-1-9 - BYE with Digest Authentication without qop

### [NAME]

UA-6-1-9 - BYE with Digest Authentication without qop

### [PURPOSE]

Verify that a NUT properly processes a BYE request with digest authentication without qop parameter.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

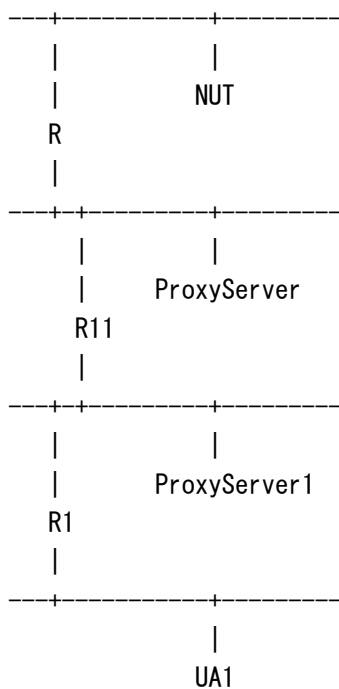
# [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# [TOPOLOGY]

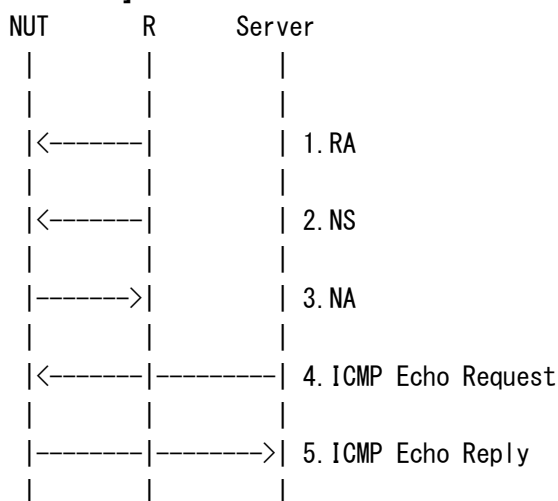


# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com

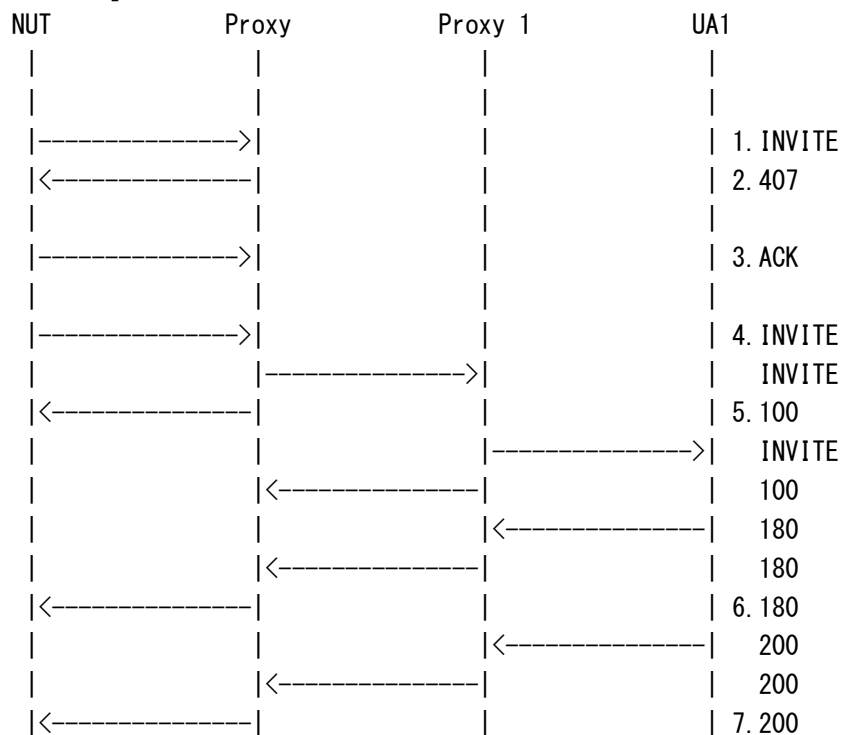
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)
-------------	--------------------------------

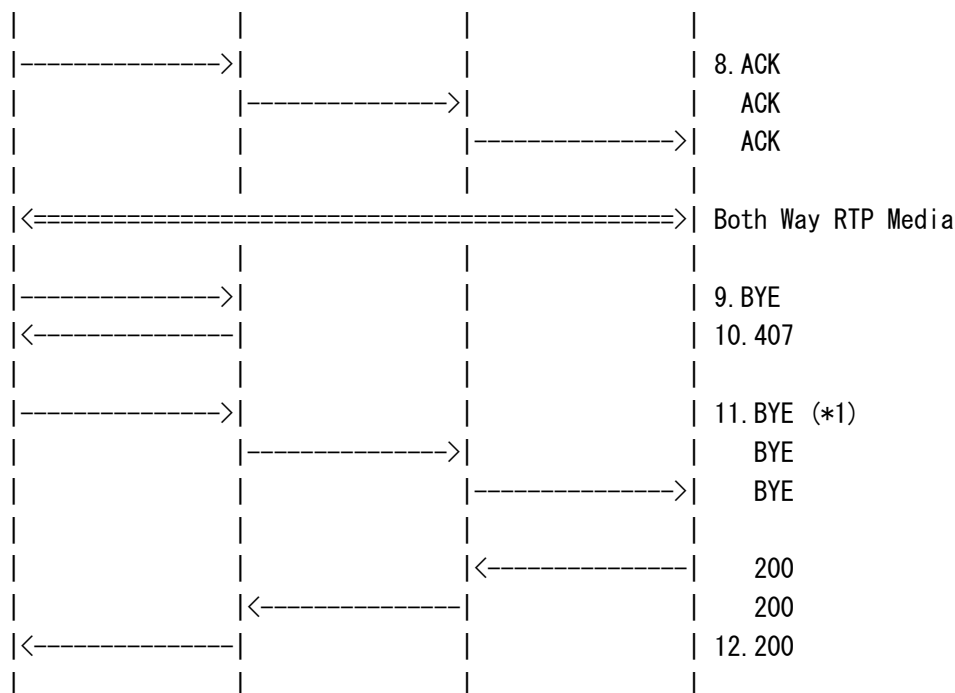
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required  
(without qop parameter in Proxy-Authenticate).
3. Receive ACK.
4. Receive INVITE (without qop parameter in Proxy-Authenticate).
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Receive BYE.
10. Send 407 Proxy Authorization Required  
(without qop parameter in Proxy-Authenticate).
11. Receive BYE (without qop parameter in Proxy-Authentication). (\*1)
12. Send 200 OK.

\* 9-10 messages are optional, so you can send 11.BYE directly.

## 2. 407 Proxy Authorization Required Proxy -> NUT

(snip)

Proxy-Authenticate: Digest realm="under.test.com",  
 nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
 stale=FALSE, algorithm=MD5



(snip)

\*without qop parameter.

#### 4. INVITE NUT -> Proxy

(snip)

Proxy-Authorization: Digest username="NUT", realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:UA1@atlanta.example.com",  
response="af8098bf32569338c4eadbf7cc3fa3be"

(snip)

#### 8. ACK NUT -> Proxy

(snip)

Proxy-Authorization: Digest username="NUT", realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:UA1@atlanta.example.com",  
response="af8098bf32569338c4eadbf7cc3fa3be"

(snip)

#### 10. 407 Proxy Authorization Required Proxy -> NUT

(snip)

Proxy-Authenticate: Digest realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
stale=FALSE, algorithm=MD5

(snip)

#### 11. BYE NUT-> Proxy

(snip)

Proxy-Authorization: Digest username="NUT", realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
uri="sip:UA1@atlanta.example.com",  
response="4767ead078938ad80e7b3a49defdc64"

(snip)

### [OBSERVABLE RESULTS]

\*1:BYE request from NUT.

As a SIP Message,



See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in 200 response. [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the To header field of "7.200" response.  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

- Bodies:

See generic\_BYE

## [REFERENCE]

[RFC3261-22-35, 36, 37, 38]

22.4 The Digest Authentication Scheme [Page 200]

8. RFC 2617 notes that a cnonce value MUST NOT be sent in an Authorization (and by extension Proxy-Authorization) header field if no qop directive has been sent. Therefore, any algorithms that have a dependency on the cnonce (including "MD5-Sess") require that the qop directive be sent. Use of



the "qop" parameter is optional in RFC 2617 for the purposes of backwards compatibility with RFC 2069; since RFC 2543 was based on RFC 2069, the "qop" parameter must unfortunately remain optional for clients and servers to receive. However, servers **MUST** always send a "qop" parameter in WWW-Authenticate and Proxy-Authenticate header field values. If a client receives a "qop" parameter in a challenge header field, it **MUST** send the "qop" parameter in any resulting authorization header field.

RFC 2543 did not allow usage of the Authentication-Info header field (it effectively used RFC 2069). However, we now allow usage of this header field, since it provides integrity checks over the bodies and provides mutual authentication. RFC 2617 [17] defines mechanisms for backwards compatibility using the qop attribute in the request. These mechanisms **MUST** be used by a server to determine if the client supports the new mechanisms in RFC 2617 that were not specified in RFC 2069.

[RFC2617-3-4, 5, 6, 7]

### 3.2.2 The Authorization Request Header field [Page 12]

#### qop

Indicates what "quality of protection" the client has applied to the message. If present, its value **MUST** be one of the alternatives the server indicated it supports in the WWW-Authenticate header field. These values affect the computation of the request-digest. Note that this is a single token, not a quoted list of alternatives as in WWW-Authenticate. This directive is optional in order to preserve backward compatibility with a minimal implementation of RFC 2069 [6], but **SHOULD** be used if the server indicated that qop is supported by providing a qop directive in the WWW-Authenticate header field.

#### cnonce

This **MUST** be specified if a qop directive is sent (see above), and **MUST NOT** be specified if the server did not send a qop directive in the WWW-Authenticate header field.

#### nonce-count

This **MUST** be specified if a qop directive is sent (see above), and **MUST NOT** be specified if the server did not send a qop directive in



the WWW-Authenticate header field.

## 4.6 Header field parameter

### 4.6.1 UA-7-1-1 - Timestamp header field in 100 response

#### [NAME]

UA-7-1-1 - Timestamp header field in 100 response

#### [PURPOSE]

Verify that a NUT properly processes a Timestamp header field in a 100 response.

#### [REQUIREMENT]

Only when a NUT supports Timestamp header field.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

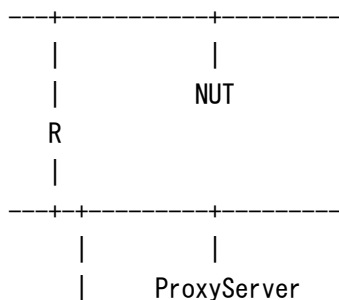
#### [PARAMETER]

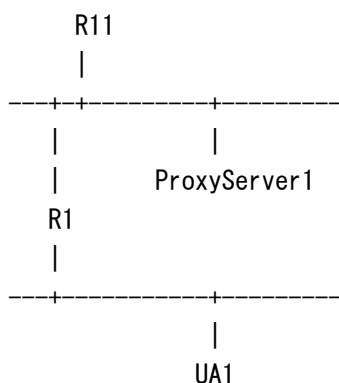
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

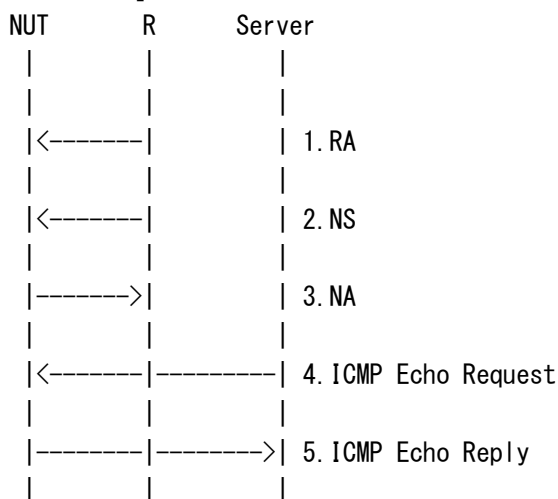




#### [CONFIGURATION for NUT]

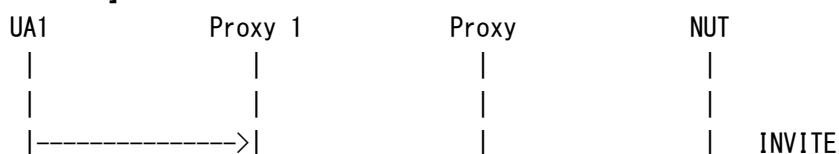
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

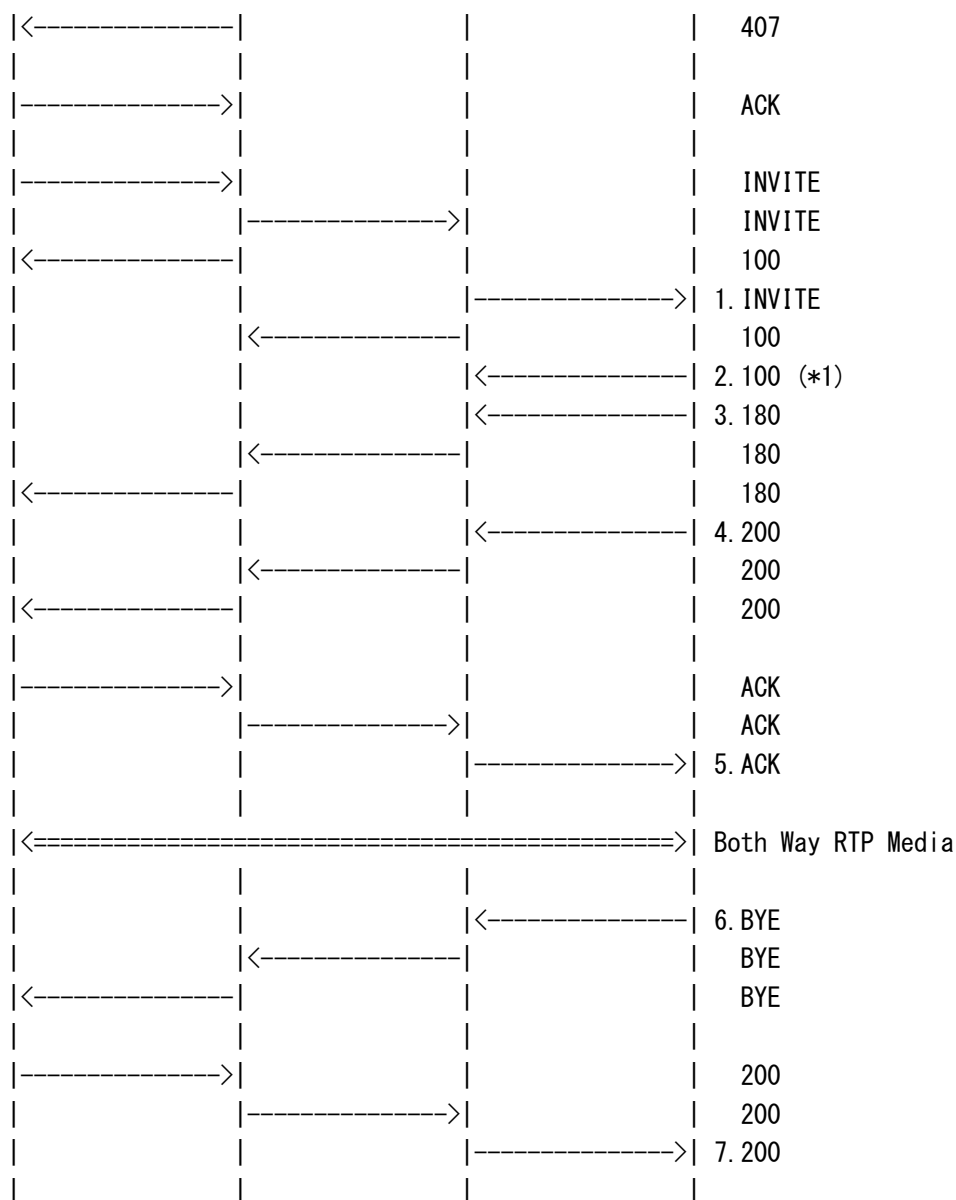
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 100 Trying. (\*1)
3. Receive 180 Ringing.
4. Receive 200 OK.
5. Send ACK.
6. Receive BYE.
7. Send 200 OK.

### 1. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0



Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 151  
Timestamp: 54

(snip)

## 2.100 Trying NUT -> Proxy

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
Contact: <sip:NUT@node.under.test.com>  
CSeq: 2 INVITE  
Content-Length: 0  
Timestamp: 54 1.5

\* "1.5" in Timestamp header field is the delay value, if there is a delay in generating the response

## [OBSERVABLE RESULTS]

\*1:100 response from NUT. [Optional]



As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Timestamp

Must exist. [RFC3261-8-95]

Must be the same as "1.INVITE"(except delay). [RFC3261-8-96]

## [REFERENCE]

[RFC3261-8-95, 96, 97]

### 8.2.6.1 Sending a Provisional Response

When a 100 (Trying) response is generated, any Timestamp header field present in the request **MUST** be copied into this 100 (Trying) response. If there is a delay in generating the response, the UAS **SHOULD** add a delay value into the Timestamp value in the response. This value **MUST** contain the difference between the time of sending of the response and receipt of the request, measured in seconds.

## 4.6.2 UA-7-1-2 - URI including a comma, question mark or semicolon

### [NAME]

UA-7-1-2 - URI including a comma, question mark or semicolon

### [PURPOSE]

Verify that a NUT properly processes a URI containing a comma, question mark, or



semicolon.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

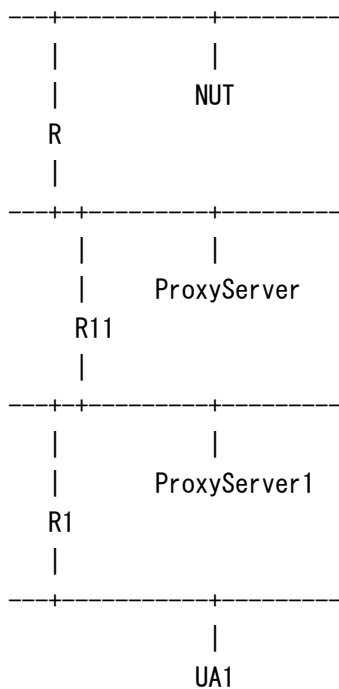
**[PARAMETER]**

NUT(AOR)	sip:NU,T@under.test.com
NUT(Contact)	sip:NU?T@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA;2@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

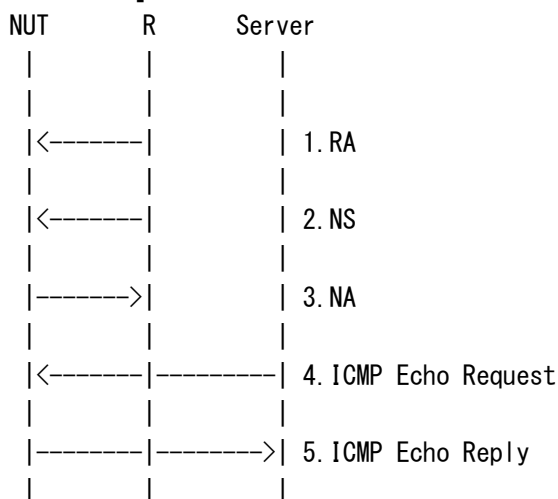
**[TOPOLOGY]**



# [CONFIGURATION for NUT]

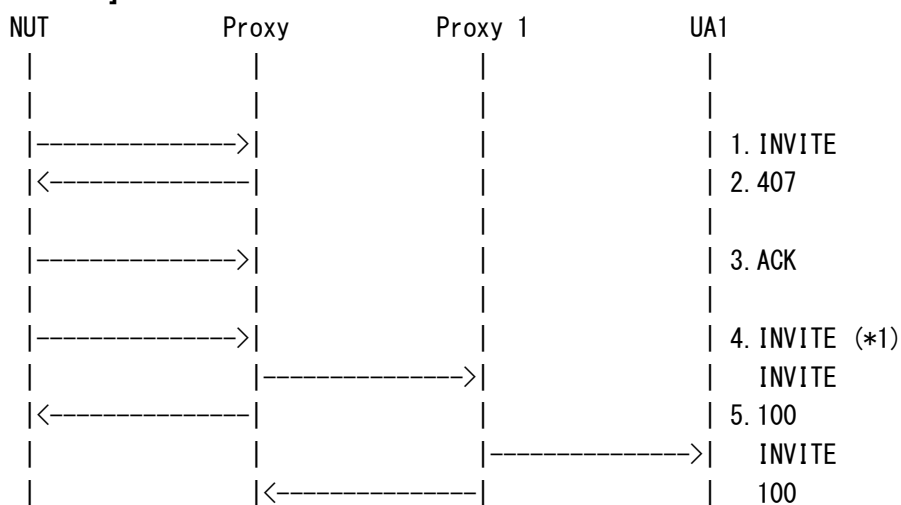
NUT(AOR)	sip:NU,T@under.test.com
NUT(Contact)	sip:NU?T@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA;2@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

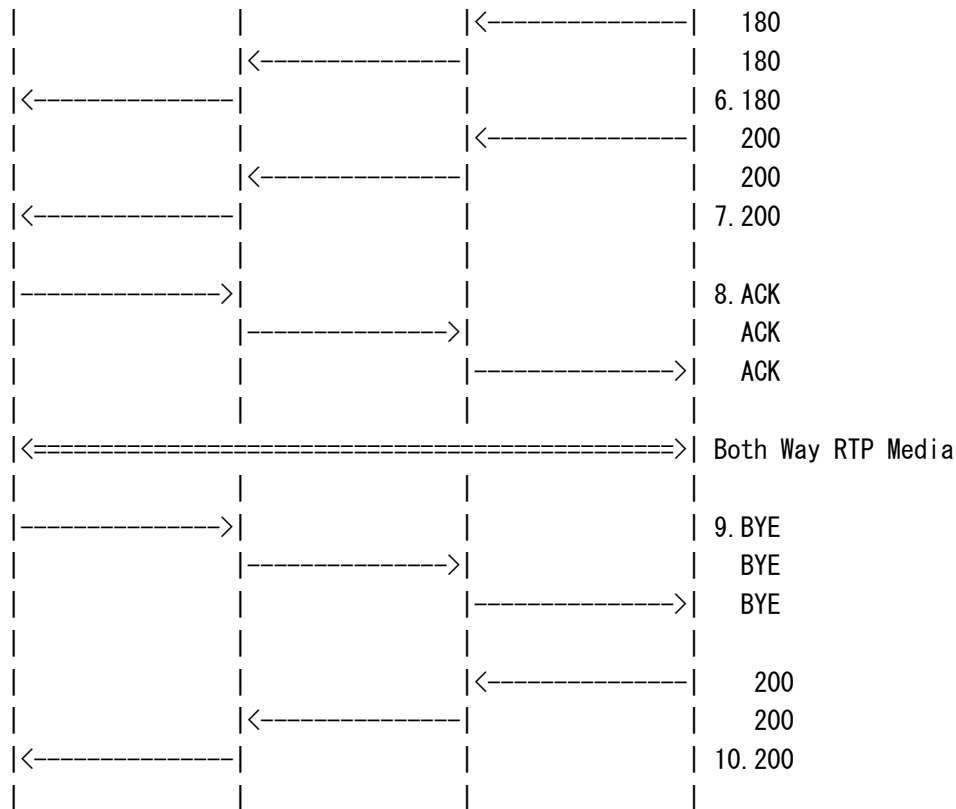
# [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE. (\*1)
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Receive BYE.
10. Send 200 OK.

\*\*\*\* See parameter

#### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,





- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE
- Header fields:  
See generic\_request
- outside of a dialog  
See generic\_Initial-INVITE
- \* Call-ID  
callid: Should be the same as that of "1.INVITE" request, with case-sensitivity on. [RFC3261-8-62]
- \* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
See generic\_digest-auth
- \* To  
addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-13]
- \* From  
addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-13]
- \* Contact  
addr-spec: Must be enclosed in "<>" if a comma, semicolon, or question mark is contained. [RFC3261-20-13]
- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

## [REFERENCE]

[RFC3261-20-13]

20 Header field Fields

The Contact, From, and To header fields contain a URI. If the URI contains a comma, question mark or semicolon, the URI MUST be enclosed in angle brackets (< and >). Any URI parameters are contained within these brackets. If the URI is not enclosed in angle brackets, any semicolon-delimited parameters are header field-parameters,

not URI parameters.

### 4.6.3 UA-7-2-1 - Receipt of BYE with an unacceptable header field

#### [NAME]

UA-7-2-1 - Receipt of BYE with an unacceptable header field

#### [PURPOSE]

Verify that a NUT properly processes when receiving a BYE request with the header field that is not acceptable.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

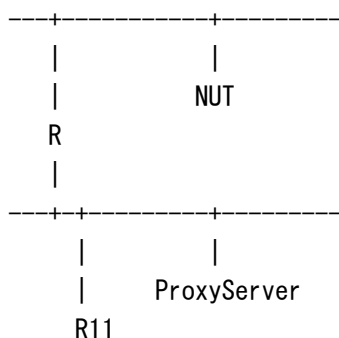
#### [PARAMETER]

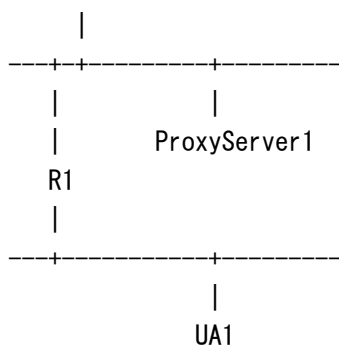
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

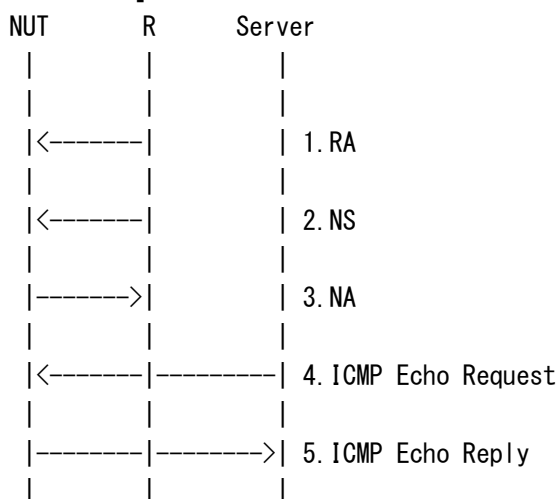




### [CONFIGURATION for NUT]

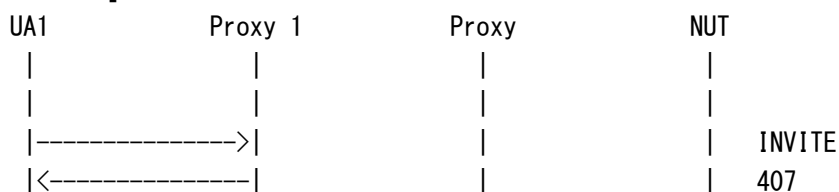
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

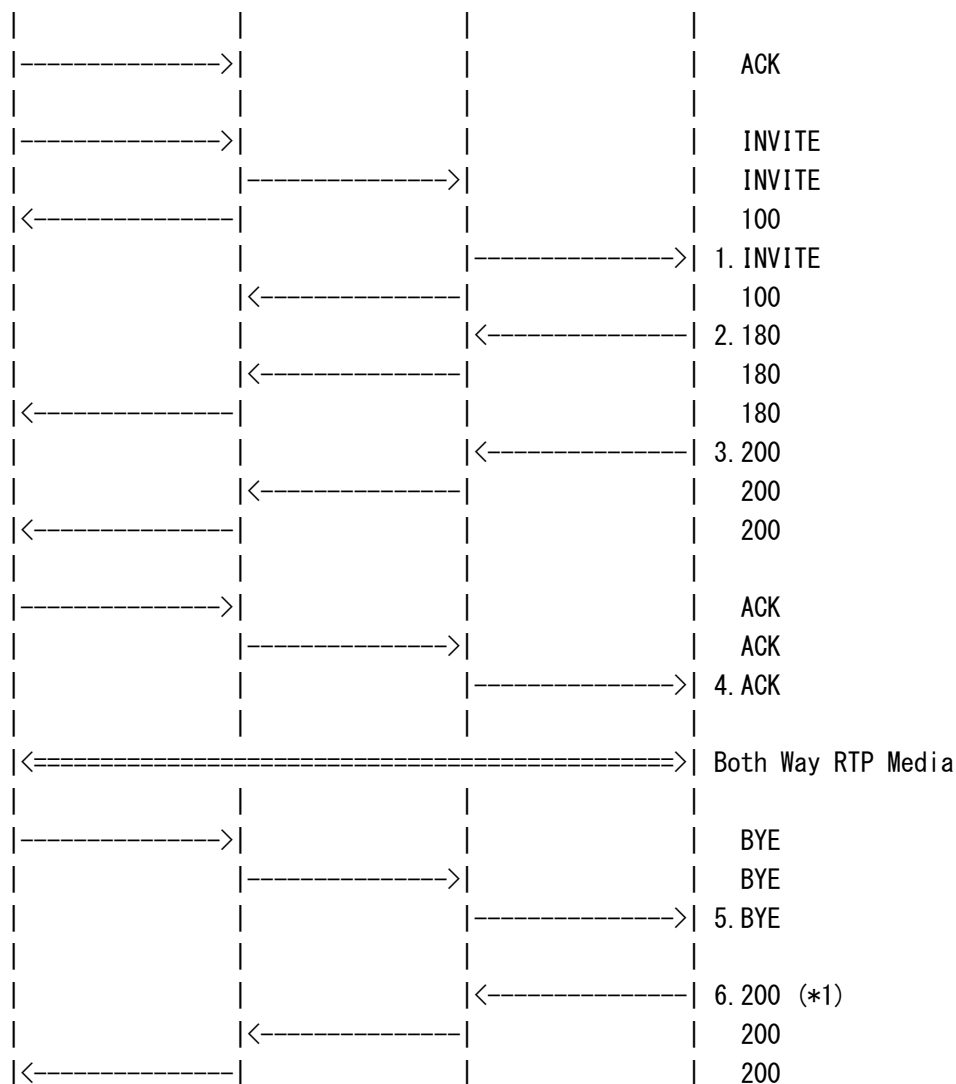
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send BYE.
6. Receive 200 OK. (\*1)

#### === Message example ===

##### 5. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK2d4790.1

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1

;received=3ffe:501:ffff:20::20



Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
Contact: <sip:UA1@client.atlanta.example.com>  
CSeq: 1 BYE  
Content-Length: 0

\* Contact header field is "Not applicable" in BYE

### **[OBSERVABLE RESULTS]**

\*1:200 response from NUT.

Must ignore invalid header field and send 200 response. [RFC3261-7-12,  
RFC3261-19-9, RFC3261-20-9]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq-2]  
Must copy all Record-Route header field values from the request into the



response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

#### [REFERENCE]

[RFC3261-20-4, 5, 6, 7, 8, 9, 10, 11]

20 Header field Fields

"Optional" means that an element MAY include the header field in a request or response, and a UA MAY ignore the header field if present in the request or response (The exception to this rule is the Require header field discussed in 20.32). A "mandatory" header field MUST be present in a request, and MUST be understood by the UAS receiving the request. A mandatory response header field MUST be present in the response, and the header field MUST be understood by the UAC processing the response. "Not applicable" means that the header field MUST NOT be present in a request. If one is placed in a request by mistake, it MUST be ignored by the UAS receiving the request. Similarly, a header field labeled "not applicable" for a response means that the UAS MUST NOT place the header field in the response, and the UAC MUST ignore the header field in the response.

#### 4.6.4 UA-7-2-2 - Receipt of CANCEL with an unacceptable header field

##### [NAME]

UA-7-2-2 - Receipt of CANCEL with an unacceptable header field

##### [PURPOSE]

Verify that a NUT properly processes when receiving CANCEL with the header field that is not acceptable.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

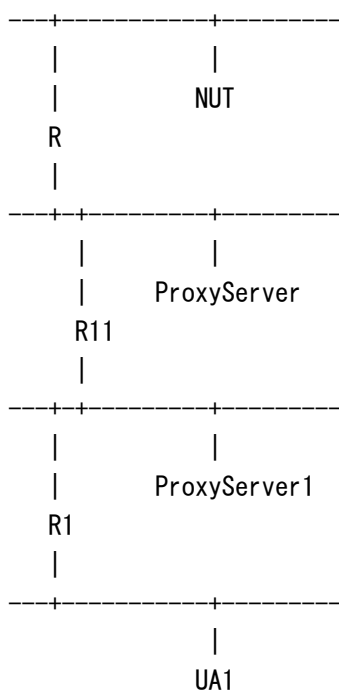
NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

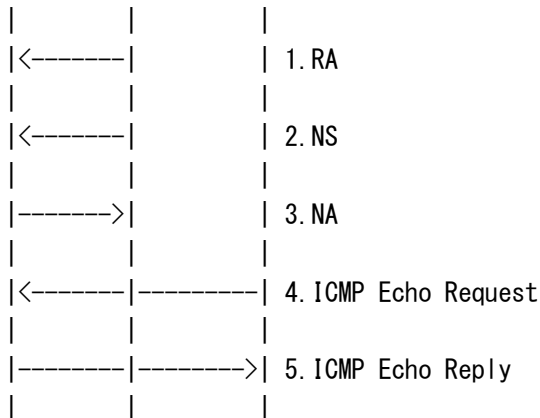


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

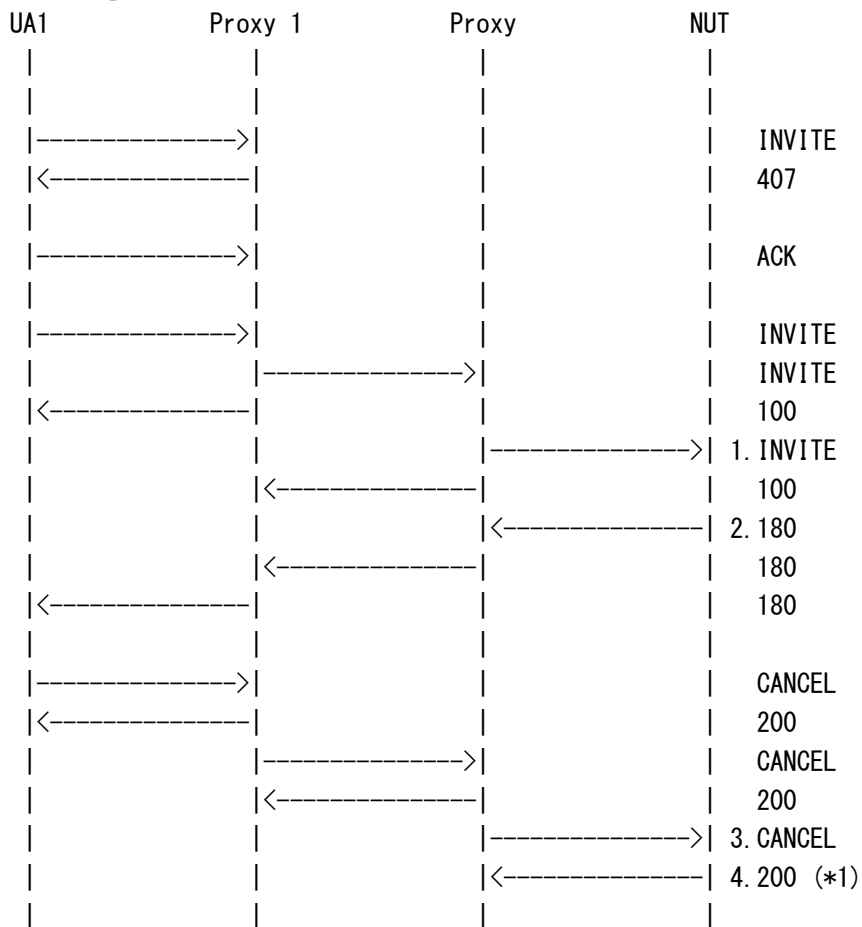
#### [INITIALIZATION]



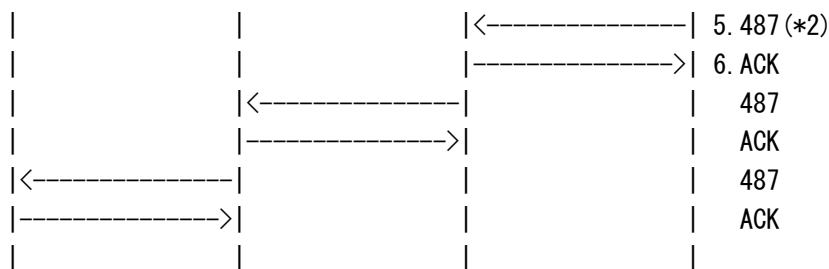


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Send INVITE.
2. Receive 180 Ringing.
3. Send CANCEL.
4. Receive 200 OK. (\*1)
5. Receive 487 Request Terminated. (\*2)
6. Send ACK.

### === Message example ===

#### 3. CANCEL NUT -> Proxy

```

CANCEL sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Max-Forwards: 70
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com
Contact: <sip:UA1@client.atlanta.example.com>
CSeq: 1 CANCEL
Content-Length: 0

```

\* Contact header field is "Not applicable" in CANCEL

### [OBSERVABLE RESULTS]

\*1:200 response from NUT.

Must ignore invalid header field and send 200 response. [RFC3261-7-12, RFC3261-19-9, RFC3261-20-9]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response



Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* To

tag-param: Should be the same as the To tag in the response to the original request. [RFC3261-9-16]

- Bodies:

See generic\_response

\*2:487 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "487". [RFC3261-9-15]

- Header fields:

See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

**[REFERENCE]**

[RFC3261-20-4, 5, 6, 7, 8, 9, 10, 11]

20 Header field Fields



"Optional" means that an element MAY include the header field in a request or response, and a UA MAY ignore the header field if present in the request or response (The exception to this rule is the Require header field discussed in 20.32). A "mandatory" header field MUST be present in a request, and MUST be understood by the UAS receiving the request. A mandatory response header field MUST be present in the response, and the header field MUST be understood by the UAC processing the response. "Not applicable" means that the header field MUST NOT be present in a request. If one is placed in a request by mistake, it MUST be ignored by the UAS receiving the request. Similarly, a header field labeled "not applicable" for a response means that the UAS MUST NOT place the header field in the response, and the UAC MUST ignore the header field in the response.

#### 4.6.5 UA-7-2-3 - Receipt of 200 with an unacceptable header field

##### [NAME]

UA-7-2-3 - Receipt of 200 with an unacceptable header field

##### [PURPOSE]

Verify that a NUT properly processes when receiving 200 response with the header field that is not acceptable.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

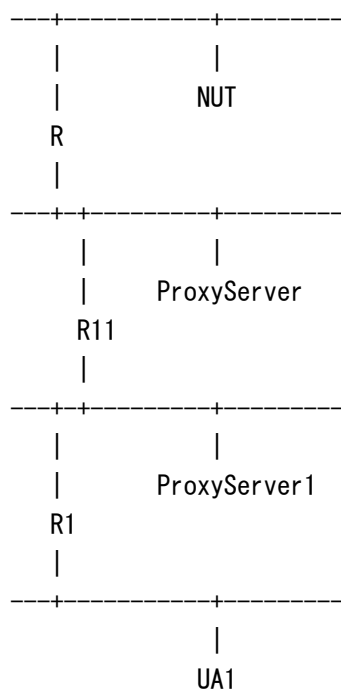
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64

ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64
--------------------	-------------------------

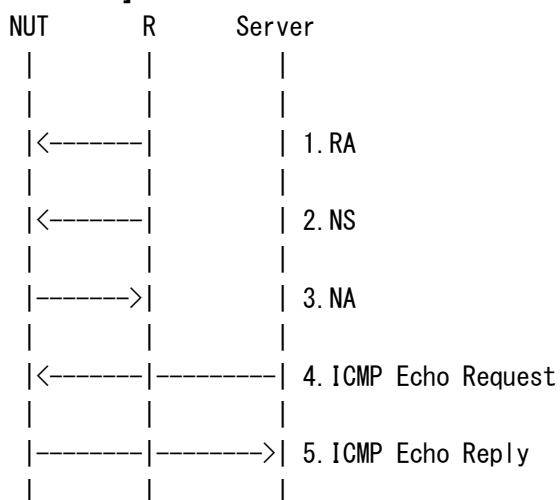
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]







1. Receive INVITE.
  2. Send 407 Proxy Authorization Required.
  3. Receive ACK.
  4. Receive INVITE.
  5. Send 100 Trying.
  6. Send 180 Ringing.
  7. Send 200 OK.
  8. Receive ACK.
  9. Receive BYE.
  10. Send 200 OK.
- (\*1)

**=== Message example ===**

**10.200 OK Proxy -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:5::X  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com  
Contact: <sip:UA1@client.atlanta.example.com>  
CSeq: 1 BYE  
Content-Length: 0

\* Contact header field is "Not applicable" in 2xx response for BYE

**[OBSERVABLE RESULTS]**

\*1:after 200 response from Proxy.

Must not send BYE request retransmission. [RFC3261-7-12, RFC3261-19-9, RFC3261-20-11]

**[REFERENCE]**

[RFC3261-20-4, 5, 6, 7, 8, 9, 10, 11]

20 Header field Fields

"Optional" means that an element MAY include the header field in a request or response, and a UA MAY ignore the header field if present in the request or response (The exception to this rule is the Require header field discussed in 20.32). A "mandatory" header field MUST be

present in a request, and MUST be understood by the UAS receiving the request. A mandatory response header field MUST be present in the response, and the header field MUST be understood by the UAC processing the response. "Not applicable" means that the header field MUST NOT be present in a request. If one is placed in a request by mistake, it MUST be ignored by the UAS receiving the request. Similarly, a header field labeled "not applicable" for a response means that the UAS MUST NOT place the header field in the response, and the UAC MUST ignore the header field in the response.

## 4.7 Routing

### 4.7.1 UA-8-1-1 - Proxy performing strict routing

#### [NAME]

UA-8-1-1 - Proxy performing strict routing

#### [PURPOSE]

Verify that a NUT properly executes processing of a strict-routing proxy.

#### [REQUIREMENT]

Only when a NUT can processes strict routing.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

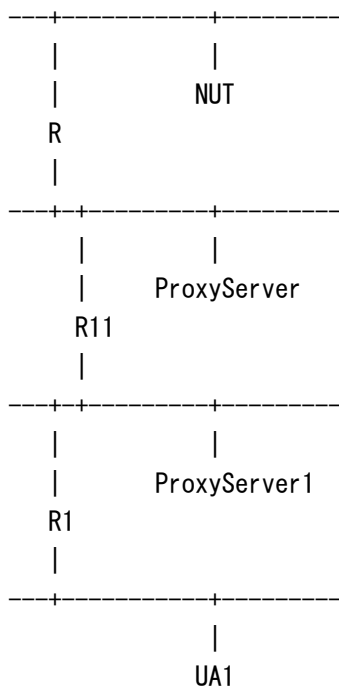
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

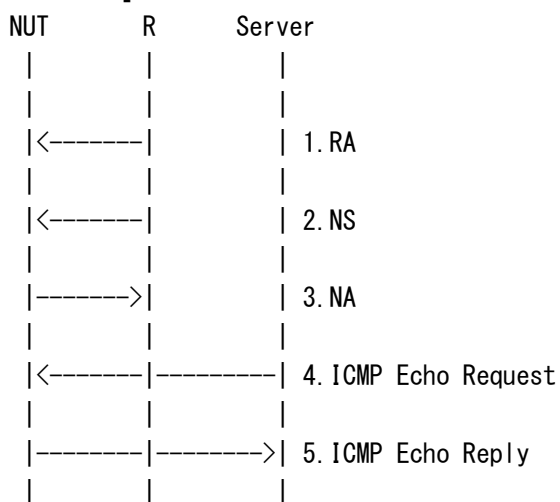
## [TOPOLOGY]



## [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

## [INITIALIZATION]

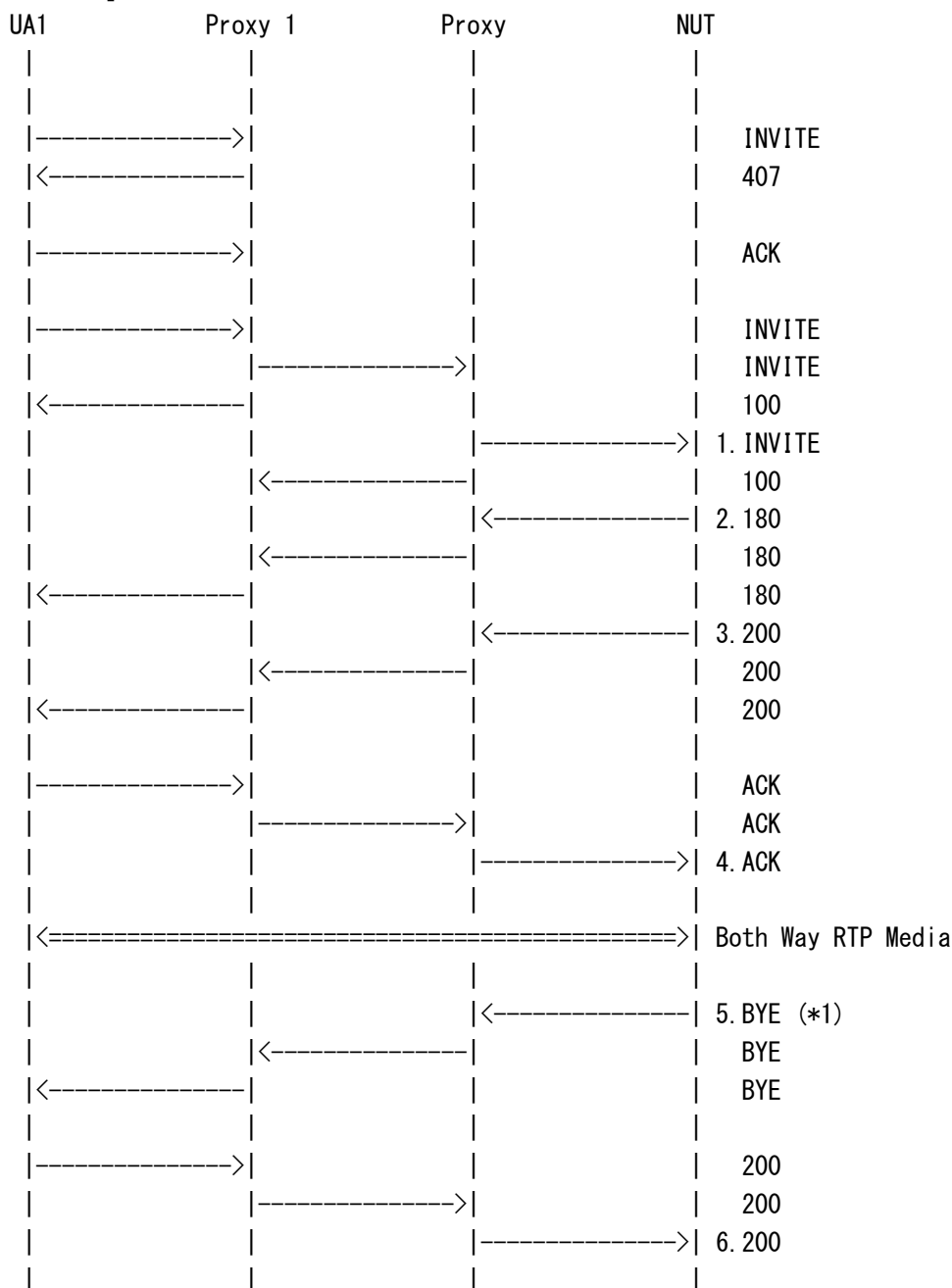


1. Send Router Advertisement.
2. Send Neighbor Solicitation.



3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]



1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.



4. Send ACK.
5. Receive BYE. (\*1)
6. Send 200 OK.

**=== Message example ===**

**1.INVITE Proxy -> NUT**

(snip)

Record-Route: <sip:ss.under.test.com>,

<sip:ss1.atlanta.example.com;lr>

Contact: <sip:UA1@client.atlanta.example.com>

(snip)

**5.BYE NUT -> Proxy**

BYE sip:ss.under.test.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7

Max-Forwards: 70

Route: <sip:ss1.atlanta.example.com;lr>,

<sip:UA1@client.atlanta.example.com>

From: NUT <sip:NUT@under.test.com>;tag=314159

To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl

Call-ID: 3848276298220188511@atlanta.example.com

CSeq: 1 BYE

Content-Length: 0

**[OBSERVABLE RESULTS]**

\*1:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request URI: Must be the URI of first URI from the route set, because that  
does not contain the lr parameter.  
[RFC3261-8-33][RFC3261-12-50]

- Header fields:

See generic\_request



- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "3.200" response. [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters, except first URI from the route set. [RFC3261-12-51]

(includes <sip:ss1.atlanta.example.com:lr>)

route-param: The last value of Route header field Must be the remote target URI. [RFC3261-12-52]

(that is, <sip:UA1@client.atlanta.example.com>)

- Bodies:

See generic\_BYE

## [REFERENCE]

[RFC3261-12-48, 50, 51]

### 12.2.1.1 Generating the Request

If the route set is not empty, and its first URI does not contain the lr parameter, the UAC MUST place the first URI from the route set into the Request-URI, stripping any parameters that are not allowed in a Request-URI. The UAC MUST add a Route header field containing the remainder of the route set values in order, including all parameters. The UAC MUST then place the remote target URI into the Route header field as the last value.

## 4.7.2 UA-8-1-2 - Detection of Merged Requests

### [NAME]

UA-8-1-2 - Detection of Merged Requests

### [PURPOSE]

Verify that a NUT properly detects merged requests.

**[REQUIREMENT]**

NONE

**[TARGET]**

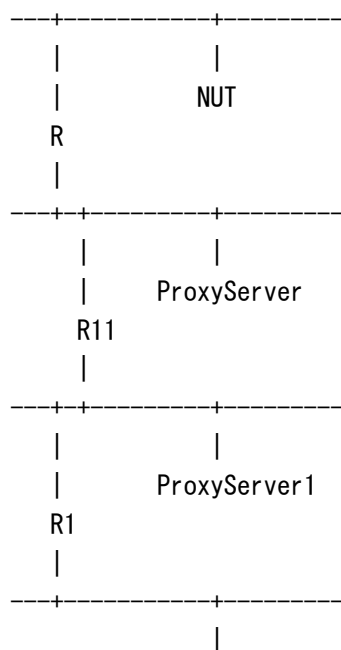
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
Proxy'	sip:ss4.under.test.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64
Proxy' (IPv6)	3ffe:501:ffff:50::51/64

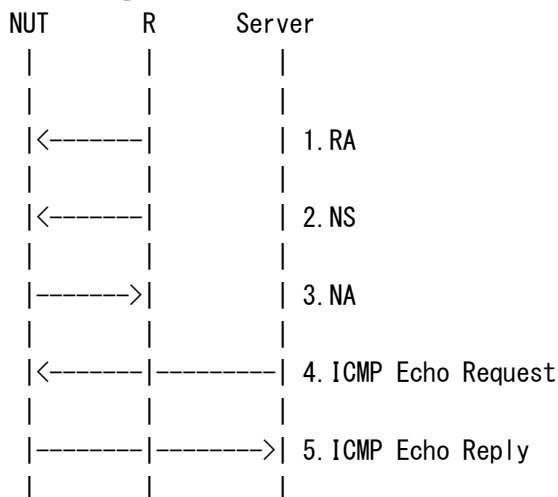
**[TOPOLOGY]**

UA1

### [CONFIGURATION for NUT]

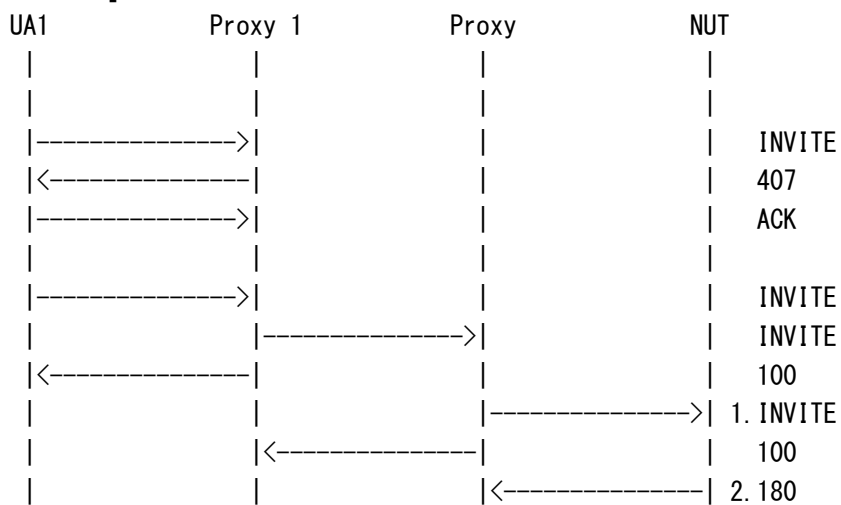
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

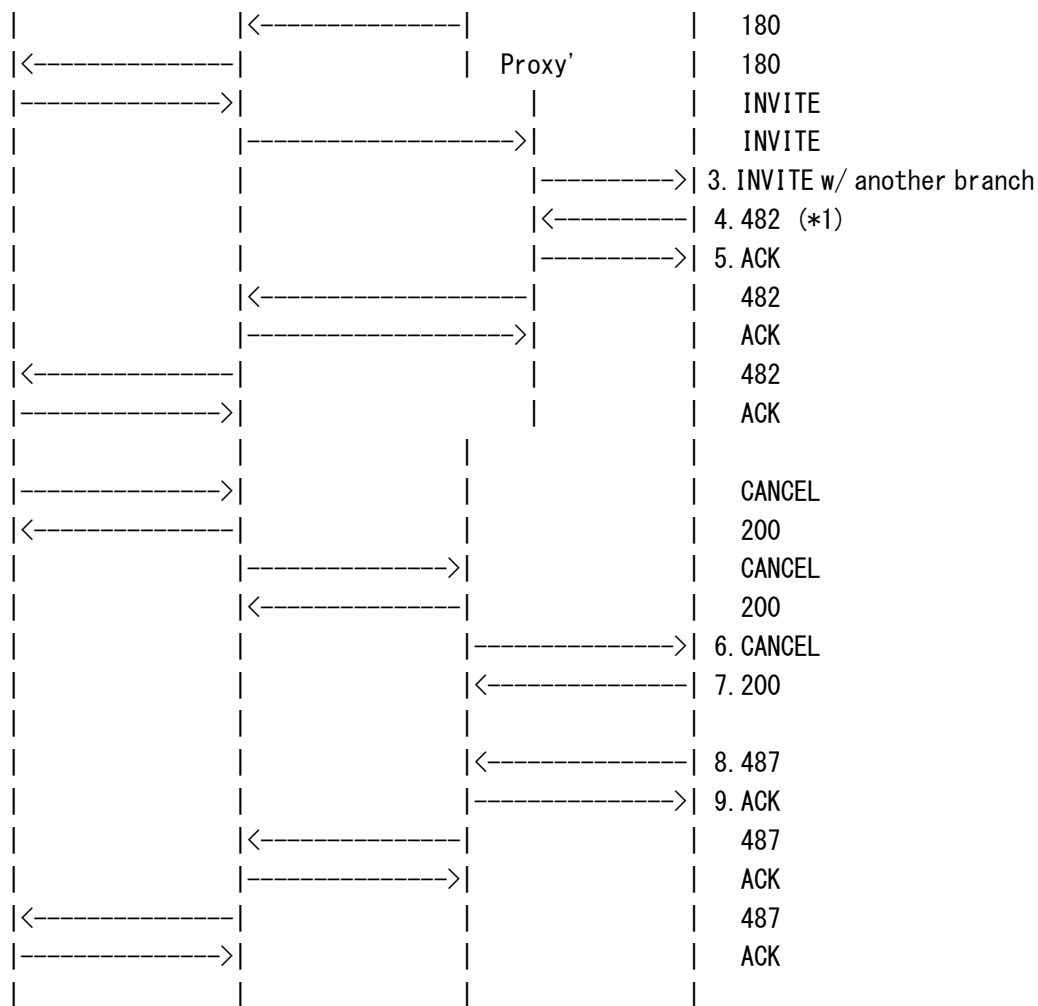
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing
3. Send INVITE.
4. Receive 482 Loop Detected. (\*1)
5. Send ACK.
6. Send CANCEL.
7. Receive 200 OK.
8. Receive 487 Request Terminated.
9. Send ACK.

#### === Message example ===

#### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
(snip)
  
```



### 3. INVITE Proxy' -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss3.under.test.com:5060;branch=z9hG4bK721e418c4.9  
(snip)

- \* The branch ID in the top Via field is different.
- \* The URI in the tompmost Via header field row is also difference.

#### [OBSERVABLE RESULTS]

\*1:482 response from NUT.

The destination address of this message Must be equal to 3ffe:501:ffff:50::51.  
[RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "482". [RFC3261-8-76, RFC3261-8-77]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

#### [REFERENCE]

[RFC3261-8-76, 77]

##### 8.2.2.2 Merged Requests

If the request has no tag in the To header field, the UAS core **MUST** check the request against ongoing transactions. If the From tag, Call-ID, and CSeq exactly match those associated with an ongoing transaction, but the request does not match that transaction (based on the matching rules in Section 17.2.3), the UAS core **SHOULD** generate a 482 (Loop Detected) response and pass it to the server

transaction.

### 4.7.3 UA-8-1-3 - "sent-by" of Via in response not inserted into request

#### [NAME]

UA-8-1-3 - "sent-by" of Via in response not inserted into a request

#### [PURPOSE]

Verify that a NUT properly processes a "sent-by" parameter in the top Via header field in a response when the NUT receives a response and the value in the parameter is not consistent with the request.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

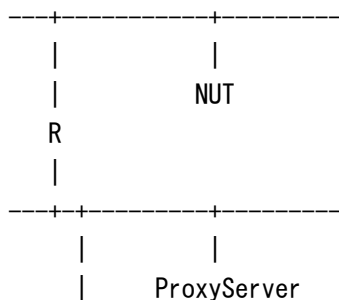
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

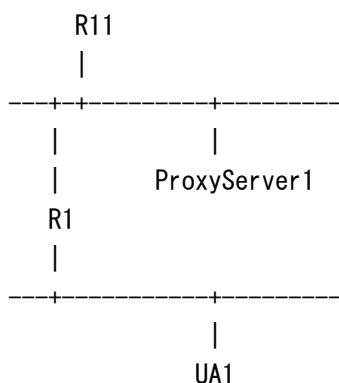
#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]



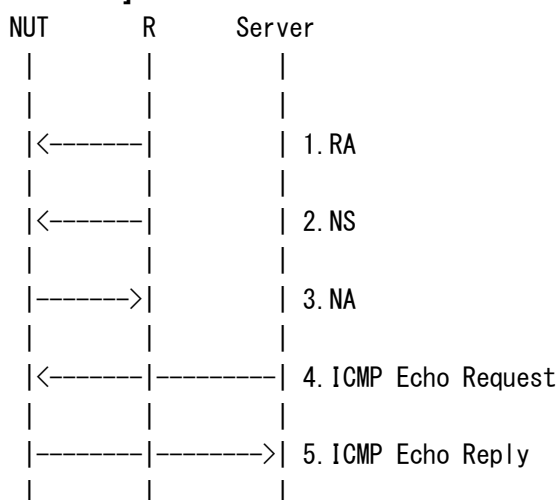




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

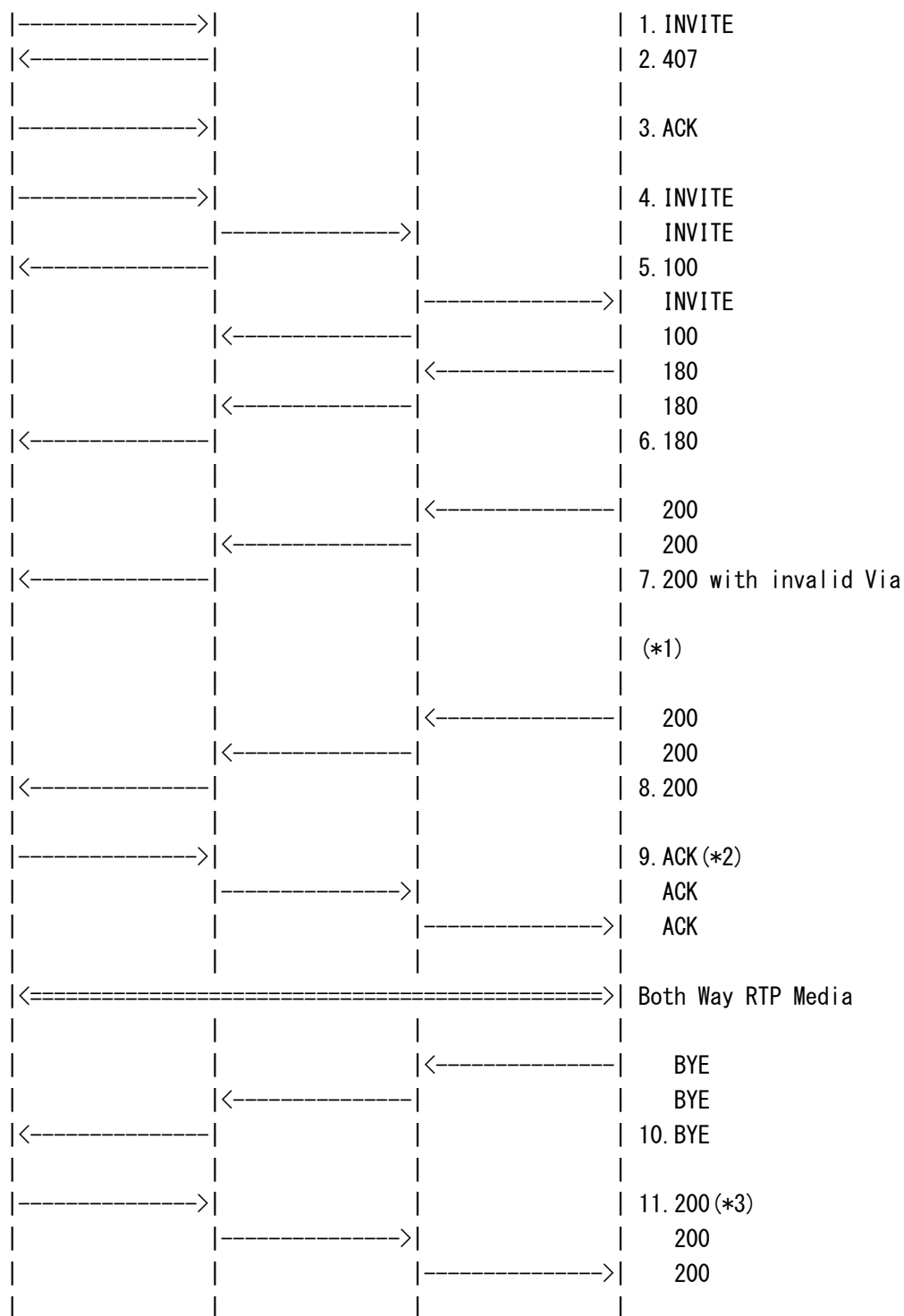
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.



5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.  
(\*1)
8. Send 200 OK.
9. Receive ACK. (\*2)
10. Send BYE.
11. Receive 200 OK. (\*3)

**=== Message example ===**

**4. INVITE NUT -> Proxy**

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
(snip)

**7. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP invalid-sent-by.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
(snip)

**8. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
(snip)

**[OBSERVABLE RESULTS]**

\*1:after 200 response from Proxy.

Must not send ACK request. [RFC3261-18-20]

\*2:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:
  - See generic\_request
  - See generic\_ACK
  - See generic\_2xx-ACK
  
- Header fields:
  - See generic\_request
  
- inside of a dialog
  - See generic\_ACK
  - See generic\_2xx-ACK
  
- \* Proxy-Authorization
  - Must exist. [RFC3261.22.3],[RFC3261-22-22]
  - Must be the same as that of Proxy-Authorization in "4.INVITE". [RFC3261-13-22]
  
- \* Route
  - Must exist. [ORq-2]
  - Must contain the Record-Route values of "8.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]
  
- Bodies:
  - See generic\_ACK
  - See generic\_2xx-ACK
  
- \*3:200 response from NUT.
  - As a SIP Message,
    - See generic\_message
  - As a SIP response,
  
- Status-Line:
  - See generic\_response
    - \* Status-Code: Must be "200". [RFC3261.22.2.1]
  
- Header fields:
  - See generic\_response
  
- inside of a dialog



**\* Via**

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

**[REFERENCE]**

[RFC3261-18-20]

**18.1.2 Receiving Responses**

When a response is received, the client transport examines the top Via header field value. If the value of the "sent-by" parameter in that header field value does not correspond to a value that the client transport is configured to insert into requests, the response MUST be silently discarded.

**4.7.4 UA-8-1-4 - "sent-by" in Via with IP address that differs from the packet source address and port**

**[NAME]**

UA-8-1-4 - "sent-by" of Via containing an IP address that differs from the packet source address and port

**[PURPOSE]**

Verify that a NUT properly processes an IP address that differs from the packet source address and a port.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

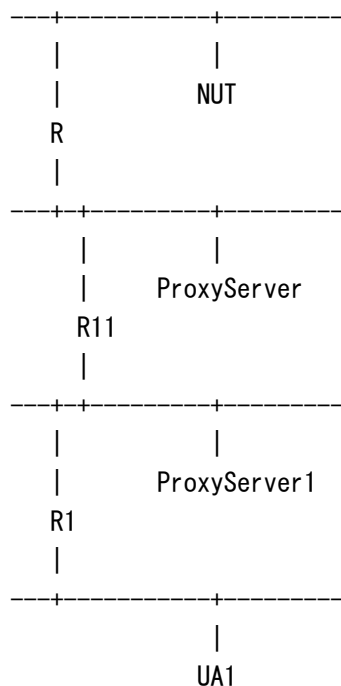
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

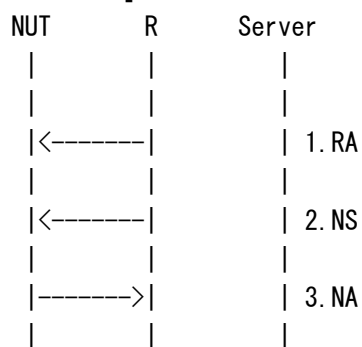
#### [TOPOLOGY]

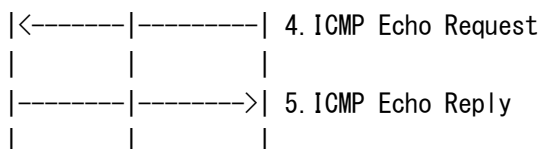


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

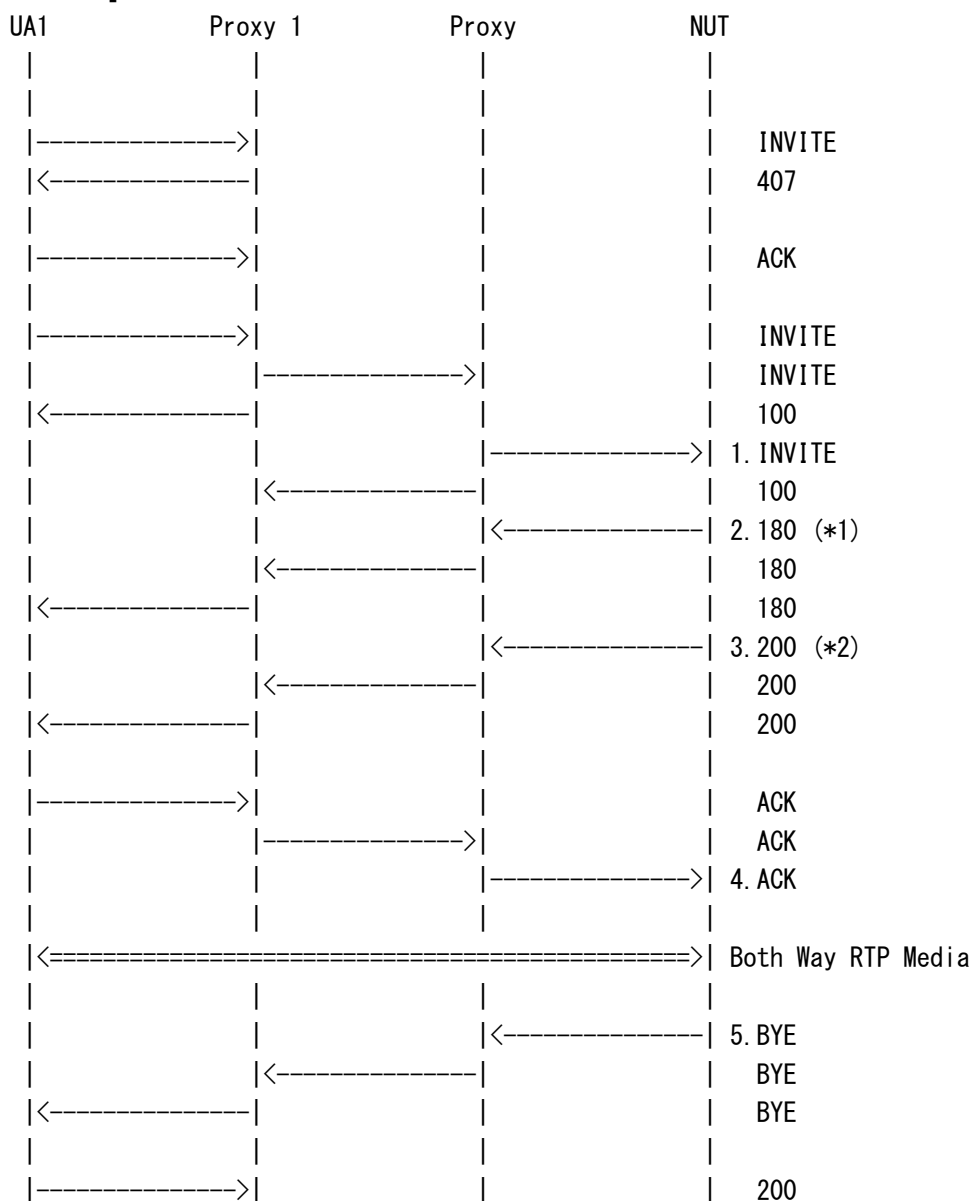
#### [INITIALIZATION]

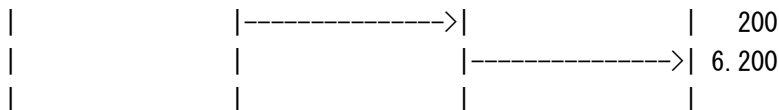




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE.
6. Send 200 OK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

```
INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP [3ffe:501:ffff:50::51]:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
* Source address = 3ffe:501:ffff:50::50
```

#### 2. 180 Ringing NUT -> Proxy

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP [3ffe:501:ffff:50::51]:5060;branch=z9hG4bK721e418c4.1
    ;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
```

#### 3. 200 OK NUT -> Proxy

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [3ffe:501:ffff:50::51]:5060;branch=z9hG4bK721e418c4.1
    ;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
```





(snip)

## **[OBSERVABLE RESULTS]**

\*1:180 response from NUT. [Optional]

The destination address of this message Must be equal to 3ffe:501:ffff:50::50.  
[RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

\*2:200 response from NUT.

The destination address of this message Must be equal to 3ffe:501:ffff:50::50.  
[RFC3261-18-35]

As a SIP Message,  
See generic\_message



As a SIP response,

- Status-Line:

See generic\_response

See generic\_200-for-INVITE

Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- outside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

## [REFERENCE]

[RFC3261-18-26, 27, 28]

### 18.2.1 Receiving Requests

When the server transport receives a request over any transport, it MUST examine the value of the "sent-by" parameter in the top Via header field value. If the host portion of the "sent-by" parameter contains a domain name, or if it contains an IP address that differs from the packet source address, the server MUST add a "received" parameter to that Via header field value. This parameter MUST contain the source address from which the packet was received. This is to assist the server transport layer in sending the response, since it must be sent to the source IP address from which the request

came.

[RFC3261-18-35, 36]

#### 18.2.2 Sending Responses

- o Otherwise (for unreliable unicast transports), if the top Via has a "received" parameter, the response MUST be sent to the address in the "received" parameter, using the port indicated in the "sent-by" value, or using port 5060 if none is specified explicitly. If this fails, for example, elicits an ICMP "port unreachable" response, the procedures of Section 5 of [4] SHOULD be used to determine where to send the response.

### 4.7.5 UA-8-1-5 - Via with "maddr" parameter and port in the "sent-by"

#### [NAME]

UA-8-1-5 - Via with "maddr" parameter and port in the "sent-by"

#### [PURPOSE]

Verify that a NUT properly processes a Via header field with a "maddr" parameter and a port in a "sent-by" parameter

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

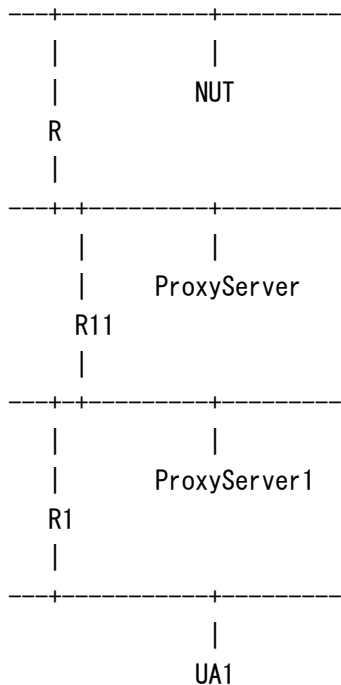
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
ProxyServer'	sip:ss2.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64

ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64
ProxyServer'(IPv6)	3ffe:501:ffff:50::51/64

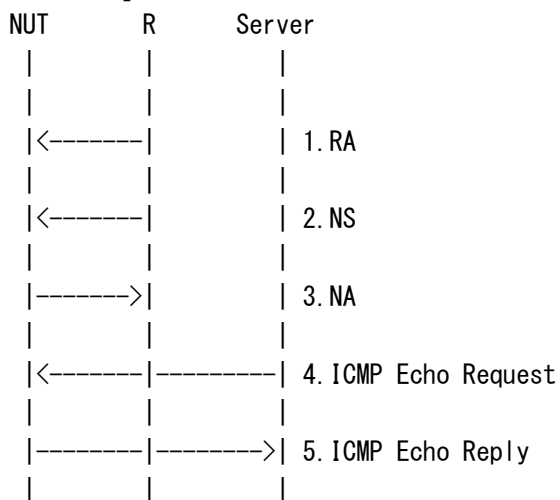
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE.
6. Send 200 OK.

**=== Message example ===**

**1. INVITE Proxy -> NUT**

```
INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5070;maddr=[3ffe:501:ffff:50::51]
    ;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
* Source address = 3ffe:501:ffff:50::50
```

**2. 180 Ringing NUT -> Proxy'**

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP ss.under.test.com:5070;maddr=[3ffe:501:ffff:50::51]
    ;branch=z9hG4bK721e418c4.1;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
```

**3. 200 OK NUT -> Proxy'**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ss.under.test.com:5070;maddr=[3ffe:501:ffff:50::51]
    ;branch=z9hG4bK721e418c4.1;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
```



#### 4.ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0

Via: SIP/2.0/UDP ss.under.test.com:5070;maddr=[3ffe:501:ffff:50::51]

;branch=z9hG4bK721e418c4.1

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9

;received=3ffe:501:ffff:1::1

(snip)

\* Source address = 3ffe:501:ffff:50::50

#### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

The destination address of this message Must be equal to 3ffe:501:ffff:50::51.  
[RFC3261-18-33]

The destination port of this message Must be equal to 5070. [RFC3261-18-33]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the  
response. [RFC3261-12-2, 9]



rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

\*2:200 response from NUT.

The destination address of this message Must be equal to 3ffe:501:ffff:50::51.  
[RFC3261-18-33]

The destination port of this message Must be equal to 5070. [RFC3261-18-33]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the  
response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP





## [REFERENCE]

[RFC3261-18-26, 27, 28]

### 18.2.1 Receiving Requests

When the server transport receives a request over any transport, it MUST examine the value of the "sent-by" parameter in the top Via header field value. If the host portion of the "sent-by" parameter contains a domain name, or if it contains an IP address that differs from the packet source address, the server MUST add a "received" parameter to that Via header field value. This parameter MUST contain the source address from which the packet was received. This is to assist the server transport layer in sending the response, since it must be sent to the source IP address from which the request came.

[RFC3261-18-35, 36]

### 18.2.2 Sending Responses

- o Otherwise, if the Via header field value contains a "maddr" parameter, the response MUST be forwarded to the address listed there, using the port indicated in "sent-by", or port 5060 if none is present. If the address is a multicast address, the response SHOULD be sent using the TTL indicated in the "ttl" parameter, or with a TTL of 1 if that parameter is not present.

## 4.7.6 UA-8-1-6 - Via with "maddr" parameter and without port in "sent-by"

### [NAME]

UA-8-1-6 - Via with "maddr" parameter and without port in the "sent-by"

### [PURPOSE]

Verify that a NUT properly processes a Via header field with "maddr" parameter and without a port in a "sent-by" parameter.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

### [PARAMETER]

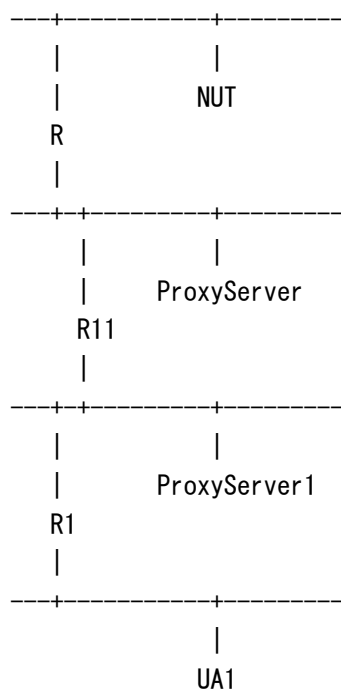
NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
ProxyServer'	sip:ss2.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64
ProxyServer'(IPv6)	3ffe:501:ffff:50::51/64

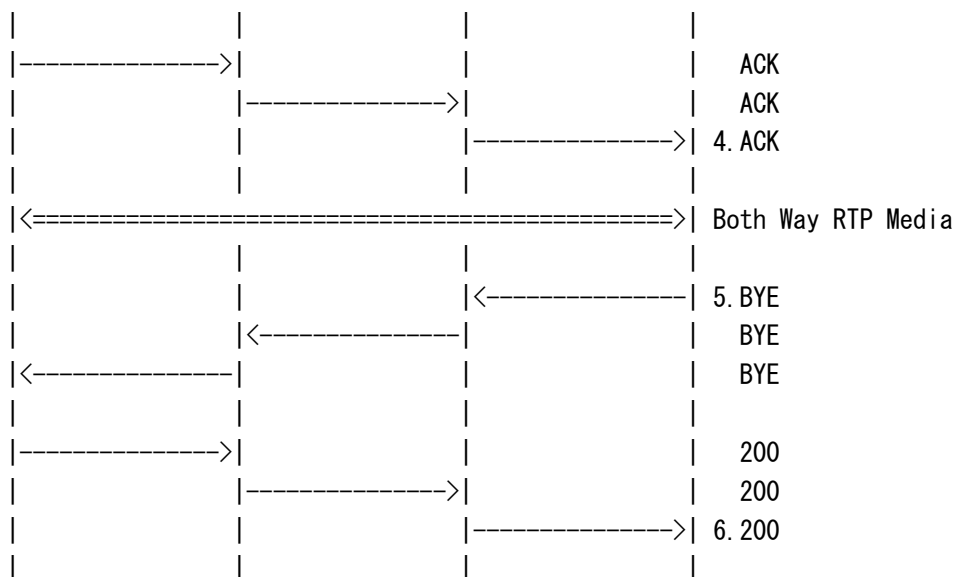
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE.
6. Send 200 OK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com;maddr=[3ffe:501:ffff:50::51]
;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
(snip)
* Source address = 3ffe:501:ffff:50::50
  
```

##### 2. 180 Ringing NUT -> Proxy'

```

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP ss.under.test.com;maddr=[3ffe:501:ffff:50::51]
;branch=z9hG4bK721e418c4.1;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
  
```



Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
(snip)

### 3.200 OK NUT -> Proxy'

SIP/2.0 200 OK

Via: SIP/2.0/UDP ss.under.test.com;maddr=[3ffe:501:ffff:50::51]  
;branch=z9hG4bK721e418c4.1;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
(snip)

### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

The destination address of this message Must be equal to 3ffe:501:ffff:50::51.  
[RFC3261-18-33]

The destination port of this message Must be equal to 5060. [RFC3261-18-33]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]



Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

\*2:200 response from NUT.

The destination address of this message Must be equal to 3ffe:501:ffff:50::51.  
[RFC3261-18-33]

The destination port of this message Must be equal to 5060. [RFC3261-18-33]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE



See generic\_SDP

#### **[REFERENCE]**

[RFC3261-18-26, 27, 28]

##### **18.2.1 Receiving Requests**

When the server transport receives a request over any transport, it **MUST** examine the value of the "sent-by" parameter in the top Via header field value. If the host portion of the "sent-by" parameter contains a domain name, or if it contains an IP address that differs from the packet source address, the server **MUST** add a "received" parameter to that Via header field value. This parameter **MUST** contain the source address from which the packet was received. This is to assist the server transport layer in sending the response, since it must be sent to the source IP address from which the request came.

[RFC3261-18-35, 36]

##### **18.2.2 Sending Responses**

- o Otherwise, if the Via header field value contains a "maddr" parameter, the response **MUST** be forwarded to the address listed there, using the port indicated in "sent-by", or port 5060 if none is present. If the address is a multicast address, the response **SHOULD** be sent using the TTL indicated in the "ttl" parameter, or with a TTL of 1 if that parameter is not present.

#### **4.7.7 UA-8-1-7 - "sent-by" in Via with a domain name and a port**

##### **[NAME]**

UA-8-1-7 - "sent-by" in Via with domain name and port

##### **[PURPOSE]**

Verify that a NUT properly processes a "sent-by" parameter in a Via header field containing a domain name and a port.

##### **[REQUIREMENT]**

NONE

##### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

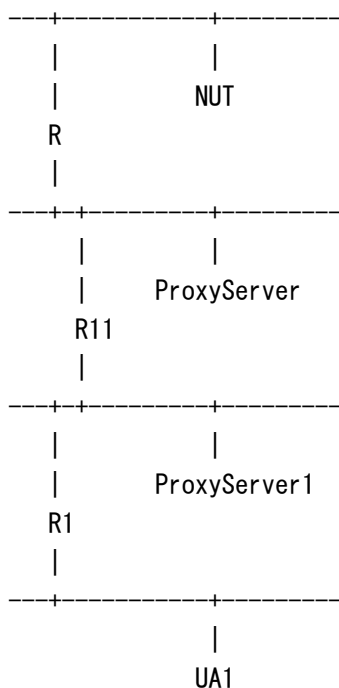
##### **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

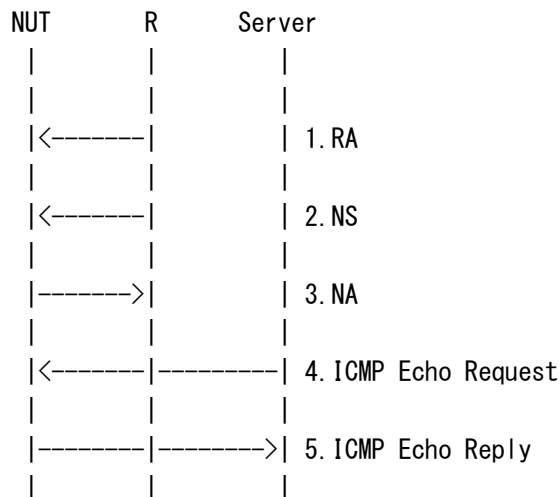


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

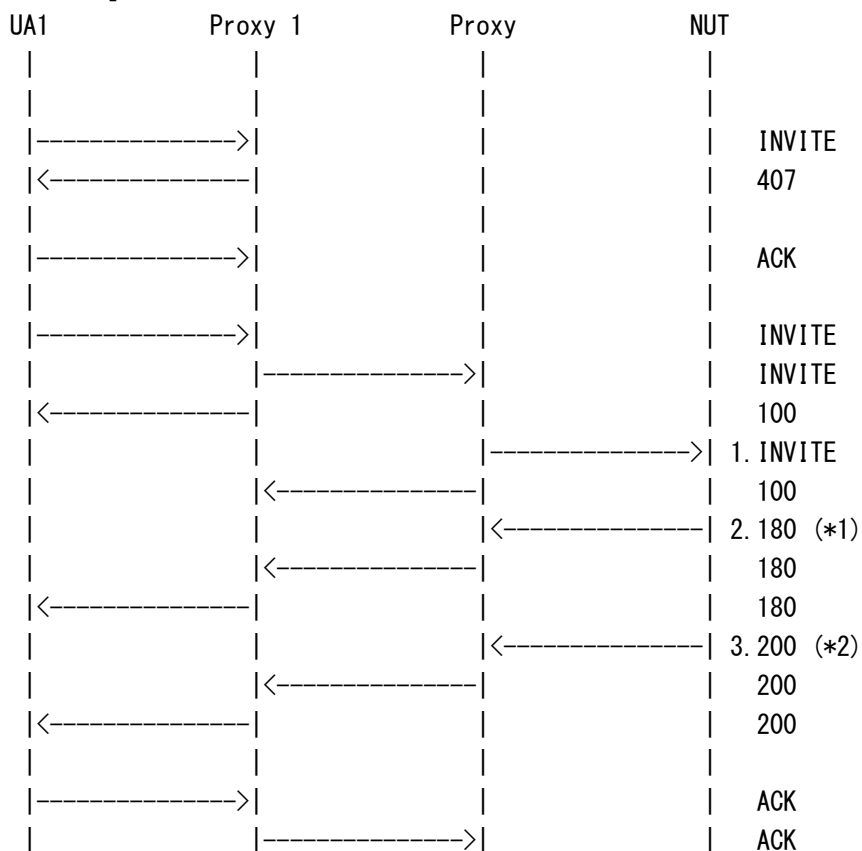
#### [INITIALIZATION]

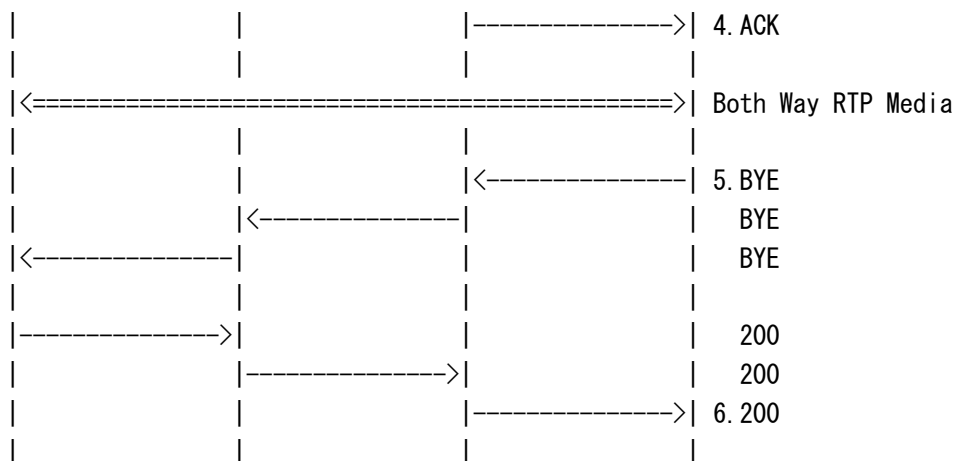




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE.
6. Send 200 OK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```
INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5061;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
(snip)
* Source address = 3ffe:501:ffff:50::50
```

##### 2. 180 Ringing NUT -> Proxy

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP ss.under.test.com:5061;branch=z9hG4bK721e418c4.1
;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
(snip)
```



### 3.200 OK NUT -> Proxy

SIP/2.0 200 OK

Via: SIP/2.0/UDP ss.under.test.com:5061;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1

(snip)

#### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

The destination address of this message Must be equal to 3ffe:501:ffff:50::50.  
[RFC3261-18-35]

The destination port of this message Must be equal to 5061. [RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the  
response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]



\*200 response from NUT.

The destination address of this message Must be equal to 3ffe:501:ffff:50::50.  
[RFC3261-18-35]

The destination port of this message Must be equal to 5061. [RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the  
response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP

## [REFERENCE]

[RFC3261-18-26, 27, 28]

### 18.2.1 Receiving Requests

When the server transport receives a request over any transport, it MUST examine the value of the "sent-by" parameter in the top Via header field value. If the host portion of the "sent-by" parameter contains a domain name, or if it contains an IP address that differs from the packet source address, the server MUST add a "received" parameter to that Via header field value. This parameter MUST contain the source address from which the packet was received. This is to assist the server transport layer in sending the response, since it must be sent to the source IP address from which the request came.

[RFC3261-18-35, 36]

### 18.2.2 Sending Responses

- o Otherwise (for unreliable unicast transports), if the top Via has a "received" parameter, the response MUST be sent to the address in the "received" parameter, using the port indicated in the "sent-by" value, or using port 5060 if none is specified explicitly. If this fails, for example, elicits an ICMP "port unreachable" response, the procedures of Section 5 of [4] SHOULD be used to determine where to send the response.

## 4.7.8 UA-8-1-8 - "sent-by" in Via with a domain name and without a port

### [NAME]

UA-8-1-8 - "sent-by" in Via with a domain name and without a port

### [PURPOSE]

Verify that a NUT properly processes a "sent-by" parameter in a Via header field containing a domain name and no port.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

### [PARAMETER]

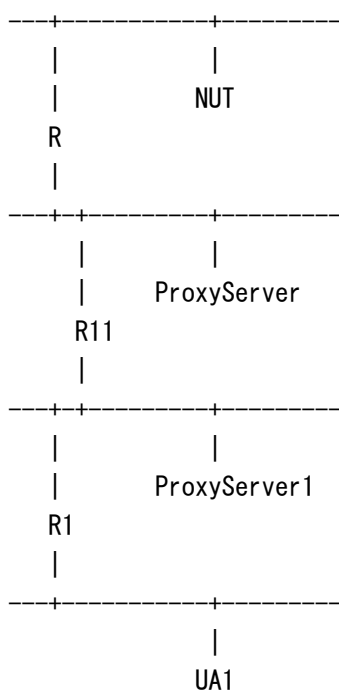
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr

UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

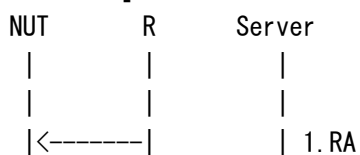
#### [TOPOLOGY]

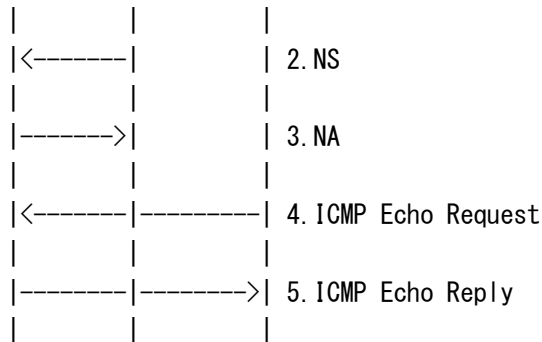


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

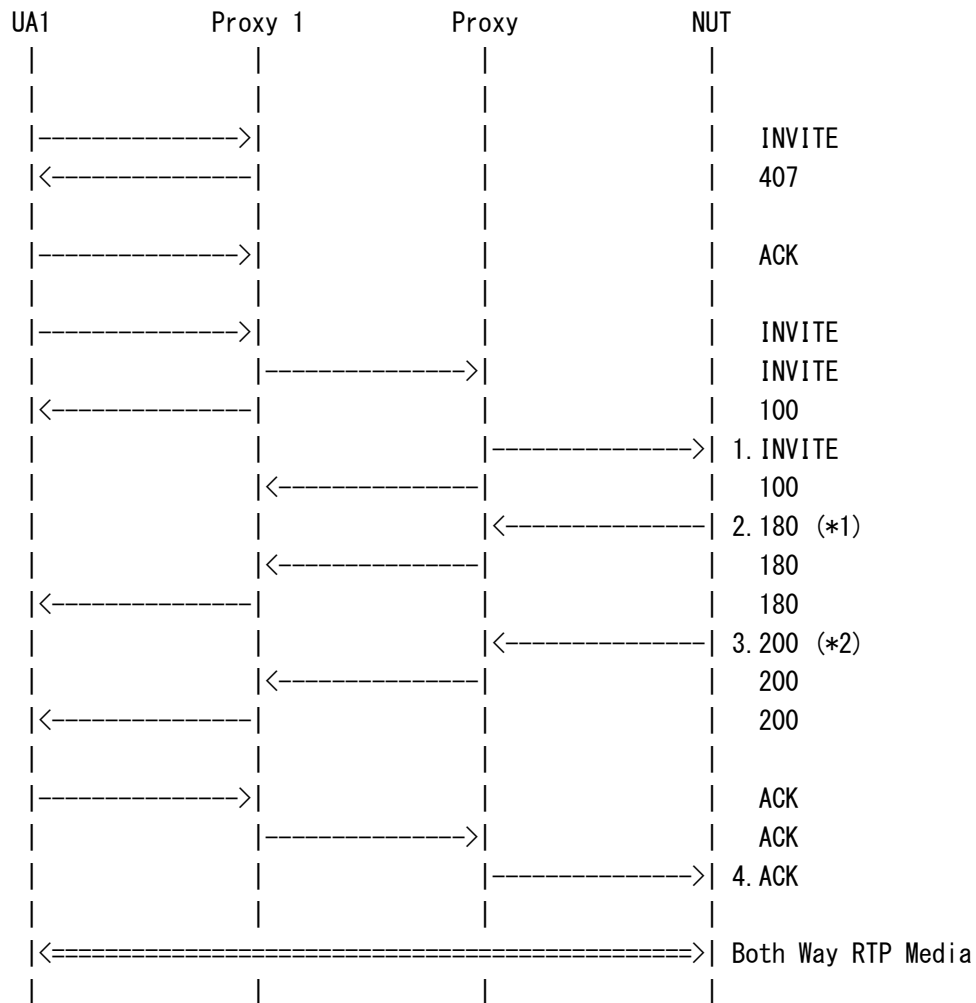
#### [INITIALIZATION]

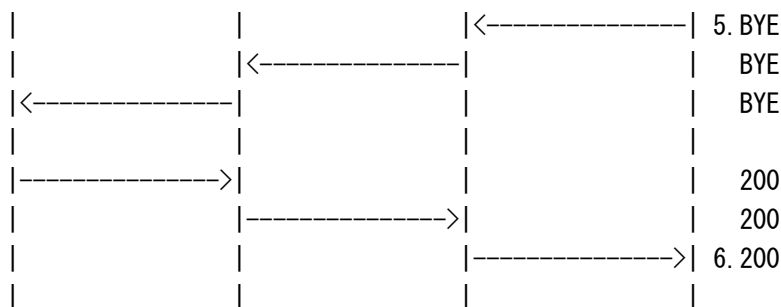




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE.
6. Send 200 OK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
* Source address = 3ffe:501:ffff:50::50
  
```

#### 2. 180 Ringing NUT -> Proxy

```

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP ss.under.test.com;branch=z9hG4bK721e418c4.1
    ;received=3ffe:501:ffff:50::50
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
(snip)
  
```

#### 3. 200 OK NUT -> Proxy

```

SIP/2.0 200 OK
Via: SIP/2.0/UDP ss.under.test.com;branch=z9hG4bK721e418c4.1
  
```





;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
(snip)

## **[OBSERVABLE RESULTS]**

\*1:180 response from NUT. [Optional]

The destination address of this message Must be equal to 3ffe:501:ffff:50::50.  
[RFC3261-18-35]

The destination port of this message Must be equal to 5060. [RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the  
response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

\*2:200 response from NUT.



The destination address of this message Must be equal to 3ffe:501:ffff:50::50. [RFC3261-18-35]

The destination port of this message Must be equal to 5060. [RFC3261-18-35]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP

## **[REFERENCE]**

[RFC3261-18-26, 27, 28]

### 18.2.1 Receiving Requests

When the server transport receives a request over any transport, it  
MUST examine the value of the "sent-by" parameter in the top Via



header field value. If the host portion of the "sent-by" parameter contains a domain name, or if it contains an IP address that differs from the packet source address, the server **MUST** add a "received" parameter to that Via header field value. This parameter **MUST** contain the source address from which the packet was received. This is to assist the server transport layer in sending the response, since it must be sent to the source IP address from which the request came.

[RFC3261-18-35, 36]

#### 18.2.2 Sending Responses

- o Otherwise (for unreliable unicast transports), if the top Via has a "received" parameter, the response **MUST** be sent to the address in the "received" parameter, using the port indicated in the "sent-by" value, or using port 5060 if none is specified explicitly. If this fails, for example, elicits an ICMP "port unreachable" response, the procedures of Section 5 of [4] **SHOULD** be used to determine where to send the response.

### 4.7.9 UA-8-1-9 - Correct Strict Routing

#### [NAME]

UA-8-1-9 - Correct Strict Routing

#### [PURPOSE]

Verify that a NUT properly processes correct strict routing.

#### [REQUIREMENT]

Only when a NUT supports strict routing.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

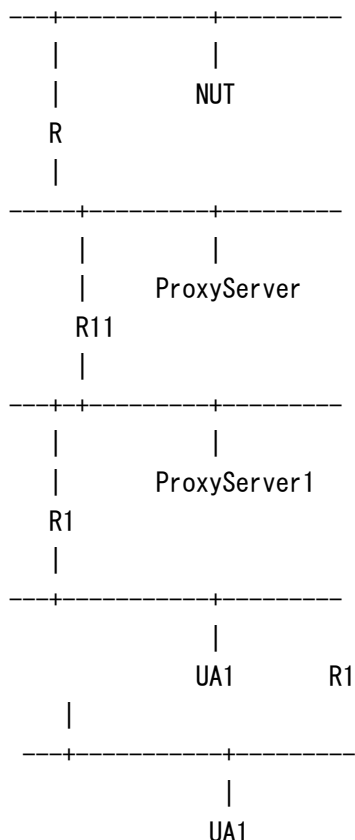
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

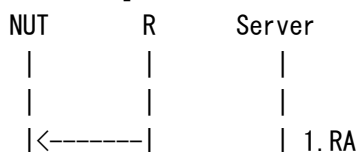
#### [TOPOLOGY]

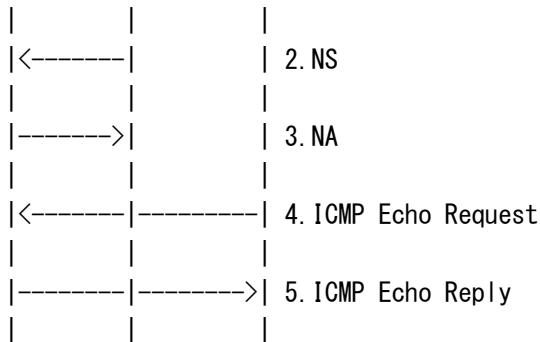


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

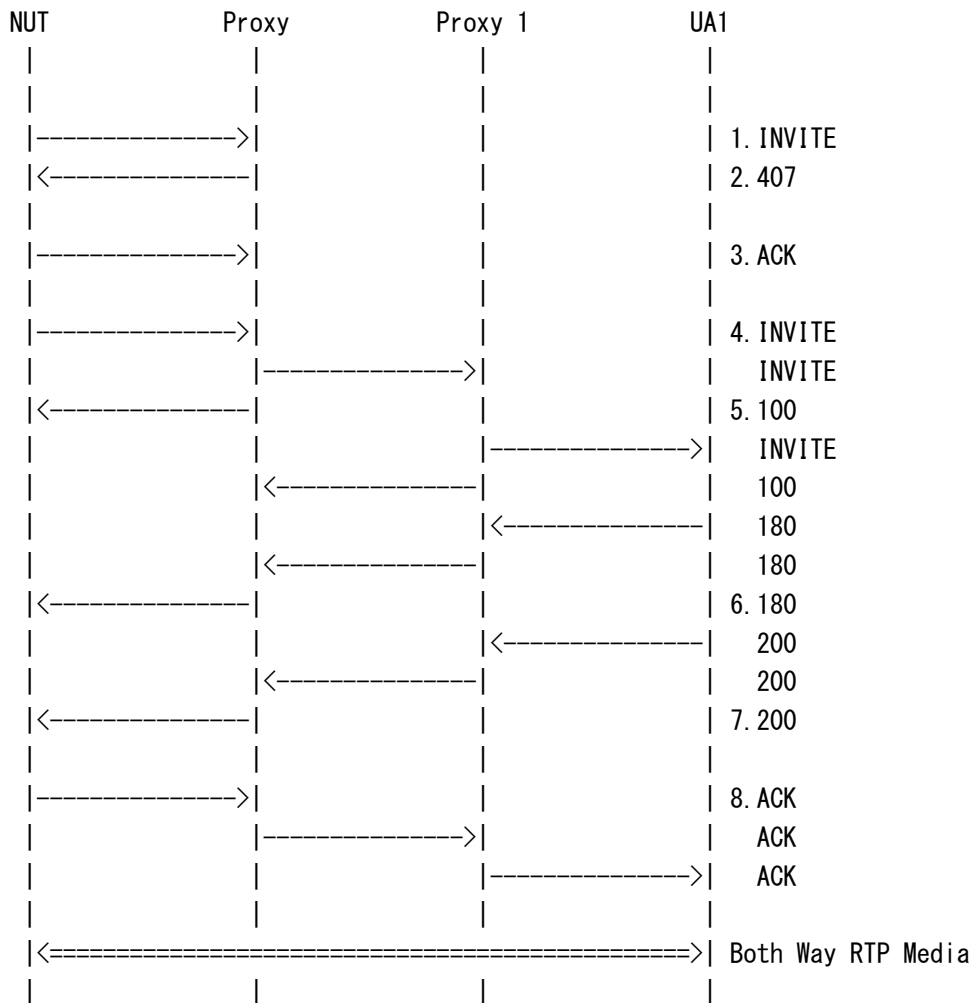
#### [INITIALIZATION]

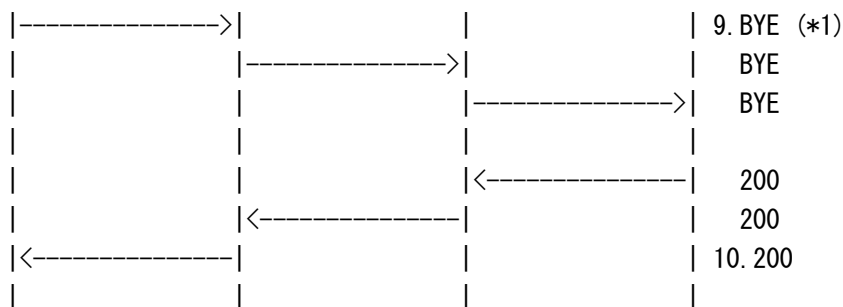




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Receive BYE. (\*1)
10. Send 200 OK.

#### === Message example ===

##### 7. 200 OK Proxy -> NUT

(snip)

Record-Route: <sip:ss1.atlanta.example.com;lr>,  
 <sip:ss.under.test.com>  
 Contact: <sip:UA1@client.atlanta.example.com>  
 (snip)

##### 9. BYE NUT -> Proxy

BYE sip:ss.under.test.com SIP/2.0  
 Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
 Max-Forwards: 70  
 Route: <sip:ss1.atlanta.example.com;lr>,  
 <sip:UA1@client.atlanta.example.com>  
 From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
 To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
 Call-ID: 3848276298220188511@under.test.com  
 CSeq: 3 BYE  
 Content-Length: 0

#### [OBSERVABLE RESULTS]



\*1:BYE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_BYE

Request URI: Must be the URI of first URI from the route set, because that does not contain the lr parameter. [RFC3261-12-50]

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_BYE

\* To  
tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-35]

\* From  
tag-param: Must equal that contained in the To header field of "3.200" response.  
[RFC3261-12-37]

\* Route  
Must include a Route header field. [RFC3261-12-48]  
route-param: Must contain the route set values in order, including all parameters, except first URI from the route set.  
[RFC3261-12-51]  
(includes <sip:ss1.atlanta.example.com;lr>)  
route-param: The last value of Route header field Must be the remote target URI. [RFC3261-12-52]  
(that is, <sip:UA1@client.atlanta.example.com>)

- Bodies:  
See generic\_BYE

#### [REFERENCE]

[RFC3261-12-50]

If the route set is not empty, and its first URI does not contain the lr parameter, the UAC MUST place the first URI from the route set

into the Request-URI, stripping any parameters that are not allowed in a Request-URI. The UAC MUST add a Route header field containing the remainder of the route set values in order, including all parameters. The UAC MUST then place the remote target URI into the Route header field as the last value.

## 4.8 Request message

### 4.8.1 UA-9-2-1 - Unrecognized type of body

#### [NAME]

UA-9-2-1 - Unrecognized type of body

#### [PURPOSE]

Verify that a NUT properly processes an unrecognized type of body.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

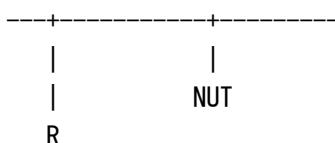
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

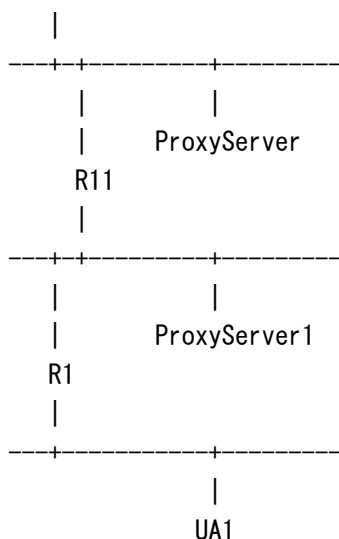
#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]



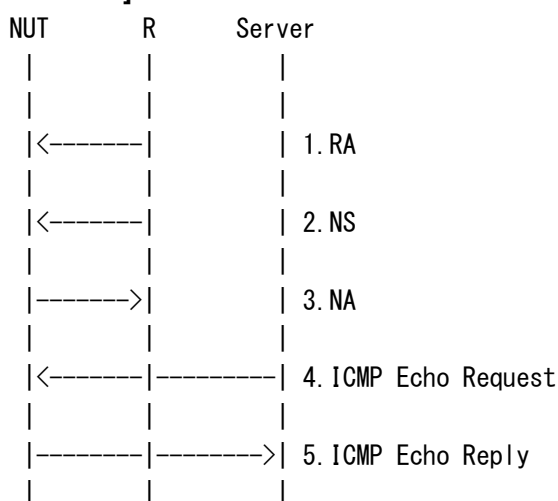




#### [CONFIGURATION for NUT]

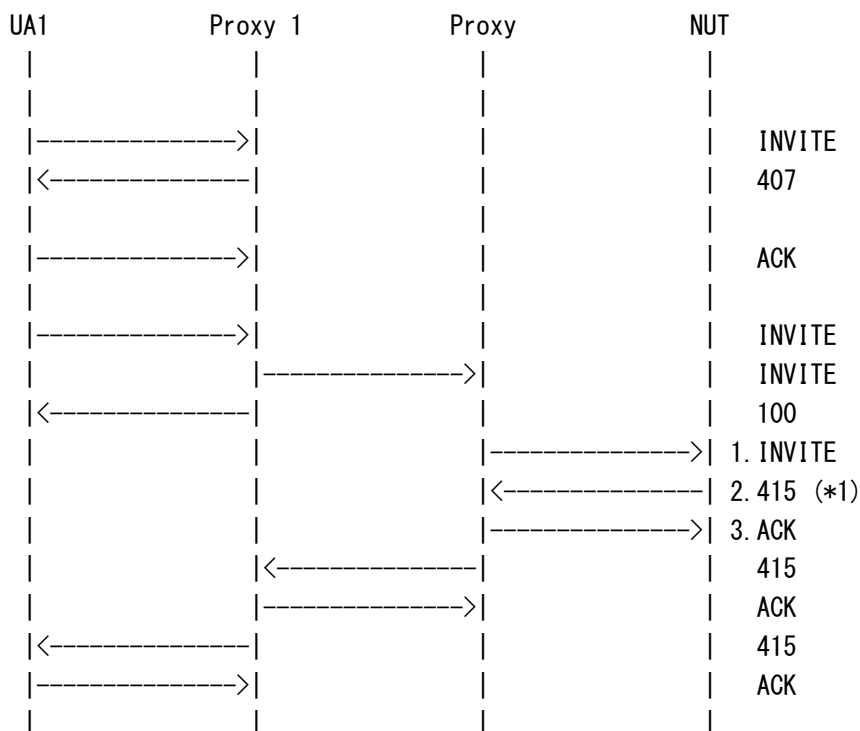
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Send INVITE.
2. Receive 415 Unsupported Media Type. (\*1)
3. Send ACK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Disposition: session;handling=required
Content-Type: unknown
  
```



Content-Length: XXX

unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown

## **[OBSERVABLE RESULTS]**

\*1:415 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "415". [RFC3261-8-83]

- Header fields:

See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Accept

Must exist. [RFC3261-21-16]

Must include the value "application/sdp".[PRq-4]

## **[REFERENCE]**

[RFC3261-8-83, 84, 85, 86]

### 8.2.3 Content Processing

Assuming the UAS understands any extensions required by the client, the UAS examines the body of the message, and the header fields that describe it. If there are any bodies whose type (indicated by the Content-Type), language (indicated by the Content-Language) or encoding (indicated by the Content-Encoding) are not understood, and that body part is not optional (as indicated by the Content-Disposition header field), the UAS MUST reject the request with a 415

(Unsupported Media Type) response. The response MUST contain an Accept header field listing the types of all bodies it understands, in the event the request contained bodies of types not supported by the UAS. If the request contained content encodings not understood by the UAS, the response MUST contain an Accept-Encoding header field listing the encodings understood by the UAS. If the request contained content with languages not understood by the UAS, the response MUST contain an Accept-Language header field indicating the languages understood by the UAS. Beyond these checks, body handling depends on the method and type. For further information on the processing of content-specific header fields, see Section 7.4 as well as Section 20.11 through 20.15.

[RFC3261-21-16]

#### 21.4.13 415 Unsupported Media Type

The server is refusing to service the request because the message body of the request is in a format not supported by the server for the requested method. The server MUST return a list of acceptable formats using the Accept, Accept-Encoding, or Accept-Language header field, depending on the specific problem with the content. UAC processing of this response is described in Section 8.1.3.5.

### 4.8.2 UA-9-2-2 - Unrecognized encoding of body

#### [NAME]

UA-9-2-2 - Unrecognized encoding of body

#### [PURPOSE]

Verify that a NUT properly unrecognized encoding of body.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com

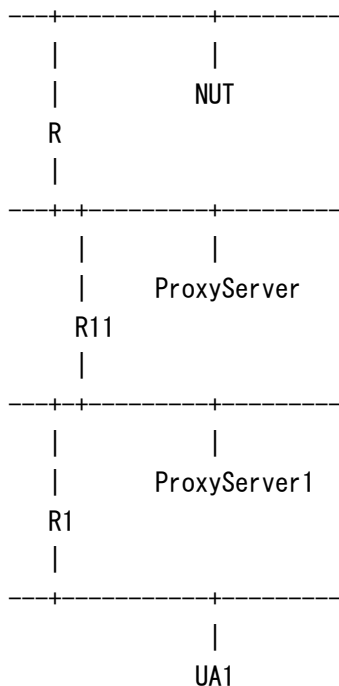


ProxyServer1	sip:ss1.atlanta.example.com;lr
--------------	--------------------------------

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

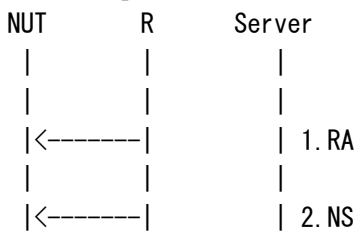
#### [TOPOLOGY]

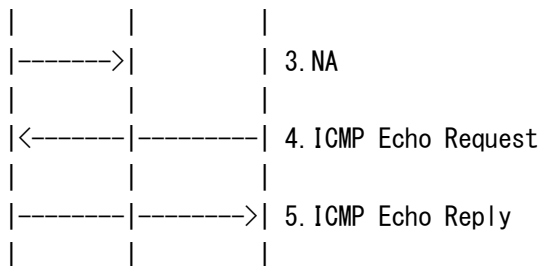


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

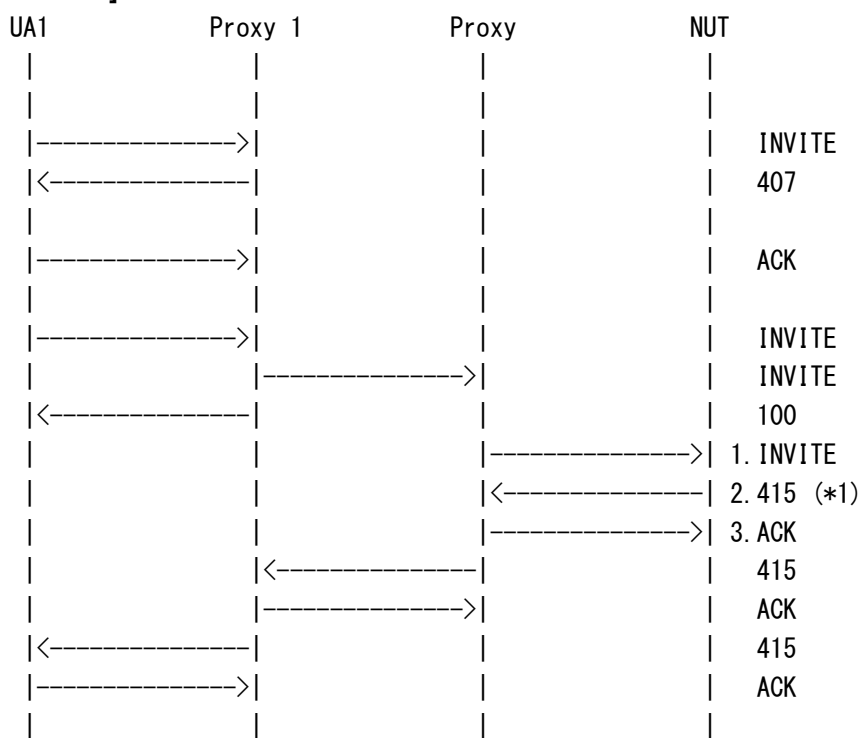
#### [INITIALIZATION]





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Send INVITE.
3. Receive 415 Unsupported Media Type. (\*1)
3. Send ACK.

=== Message example ===

1. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0



Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Disposition: session;handling=required  
Content-Encoding: unknownEncoding  
Content-Length: XXX

unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown  
unknownunknownunknownunknownunknownunknown

#### [OBSERVABLE RESULTS]

\*1:415 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "415". [RFC3261-8-83]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was  
received. [RFC3261-18-28]



\* Accept-Encoding  
Must exist. [RFC3261-21-16]

## [REFERENCE]

[RFC3261-8-83, 84, 85, 86]

### 8.2.3 Content Processing

Assuming the UAS understands any extensions required by the client, the UAS examines the body of the message, and the header fields that describe it. If there are any bodies whose type (indicated by the Content-Type), language (indicated by the Content-Language) or encoding (indicated by the Content-Encoding) are not understood, and that body part is not optional (as indicated by the Content-Disposition header field), the UAS MUST reject the request with a 415 (Unsupported Media Type) response. The response MUST contain an Accept header field listing the types of all bodies it understands, in the event the request contained bodies of types not supported by the UAS. If the request contained content encodings not understood by the UAS, the response MUST contain an Accept-Encoding header field listing the encodings understood by the UAS. If the request contained content with languages not understood by the UAS, the response MUST contain an Accept-Language header field indicating the languages understood by the UAS. Beyond these checks, body handling depends on the method and type. For further information on the processing of content-specific header fields, see Section 7.4 as well as Section 20.11 through 20.15.

[RFC3261-21-16]

### 21.4.13 415 Unsupported Media Type

The server is refusing to service the request because the message body of the request is in a format not supported by the server for the requested method. The server MUST return a list of acceptable formats using the Accept, Accept-Encoding, or Accept-Language header field, depending on the specific problem with the content. UAC processing of this response is described in Section 8.1.3.5.

## 4.8.3 UA-9-2-3 - Body in unrecognized language

### [NAME]

UA-9-2-3 - Body in Unrecognized language

### [PURPOSE]





Verify that a NUT properly processes a body in unrecognized language.

**[REQUIREMENT]**

NONE

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

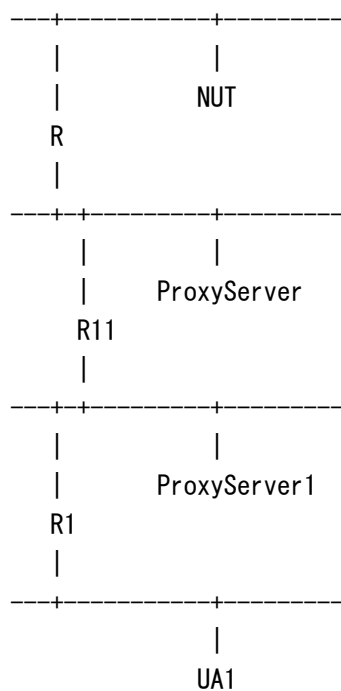
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

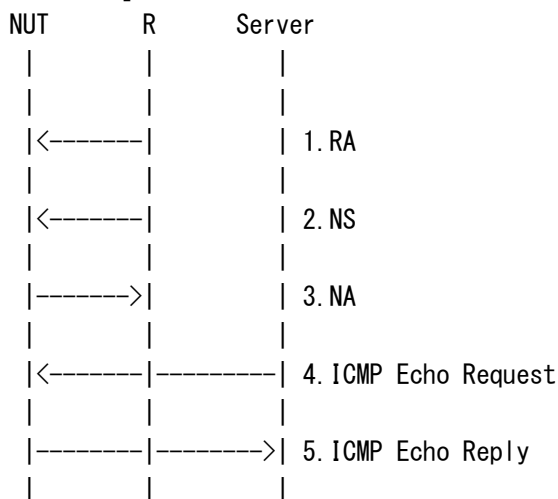
**[TOPOLOGY]**



# [CONFIGURATION for NUT]

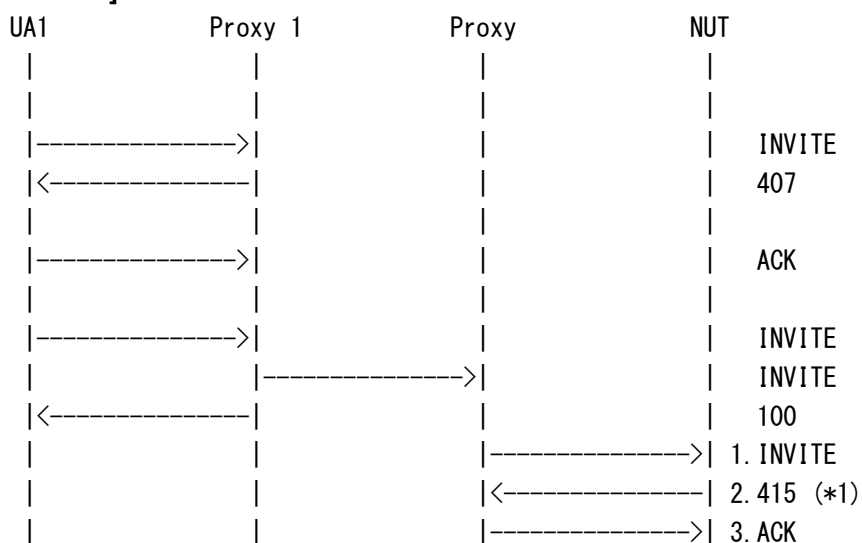
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

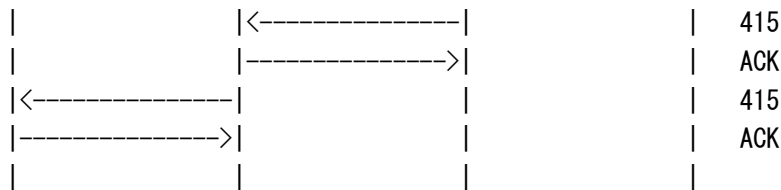
# [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]





1. Send INVITE.
2. Receive 415 Unsupported Media Type. (\*1)
3. Send ACK.

**=== Message example ===**

**1. INVITE Proxy -> NUT**

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Disposition: session;handling=required
Content-Language: unknown Language
Content-Length: XXX

```

unknownunknownunknownunknownunknownunknown  
 unknownunknownunknownunknownunknownunknown  
 unknownunknownunknownunknownunknownunknown  
 unknownunknownunknownunknownunknownunknown  
 unknownunknownunknownunknownunknownunknown

**[OBSERVABLE RESULTS]**

\*1:415 response from NUT.

As a SIP Message,  
 See generic\_message



As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "415". [RFC3261-8-83]

- Header fields:

See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Accept-Language

Must exist. [RFC3261-21-16]

## [REFERENCE]

[RFC3261-8-83, 84, 85, 86]

### 8.2.3 Content Processing

Assuming the UAS understands any extensions required by the client, the UAS examines the body of the message, and the header fields that describe it. If there are any bodies whose type (indicated by the Content-Type), language (indicated by the Content-Language) or encoding (indicated by the Content-Encoding) are not understood, and that body part is not optional (as indicated by the Content-Disposition header field), the UAS MUST reject the request with a 415 (Unsupported Media Type) response. The response MUST contain an Accept header field listing the types of all bodies it understands, in the event the request contained bodies of types not supported by the UAS. If the request contained content encodings not understood by the UAS, the response MUST contain an Accept-Encoding header field listing the encodings understood by the UAS. If the request contained content with languages not understood by the UAS, the response MUST contain an Accept-Language header field indicating the languages understood by the UAS. Beyond these checks, body handling depends on the method and type. For further information on the processing of content-specific header fields, see Section 7.4 as well as Section 20.11 through 20.15.

[RFC3261-21-16]

### 21.4.13 415 Unsupported Media Type

The server is refusing to service the request because the message body of the request is in a format not supported by the server for the requested method. The server MUST return a list of acceptable formats using the Accept, Accept-Encoding, or Accept-Language header field, depending on the specific problem with the content. UAC processing of this response is described in Section 8.1.3.5.

#### 4.8.4 UA-9-2-4 - Request with a tag in To header field

##### [NAME]

UA-9-2-4 - Request with a tag in the To header field

##### [PURPOSE]

Verify that a NUT properly processes a request with a tag in a To header field.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

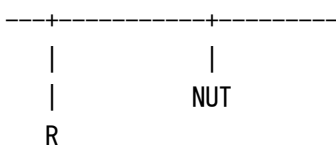
##### [PARAMETER]

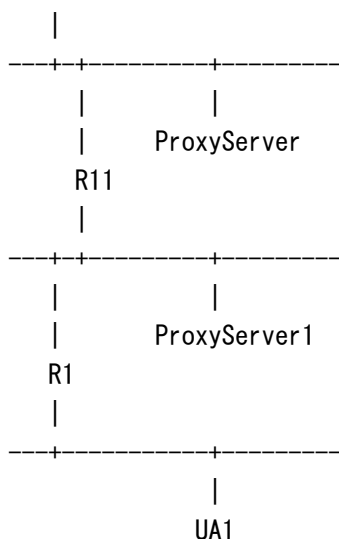
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

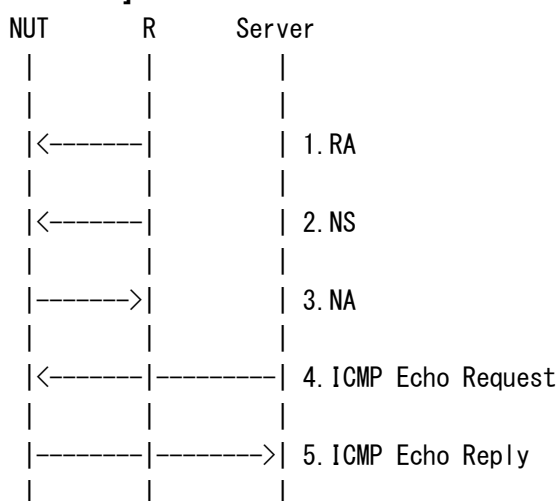




#### [CONFIGURATION for NUT]

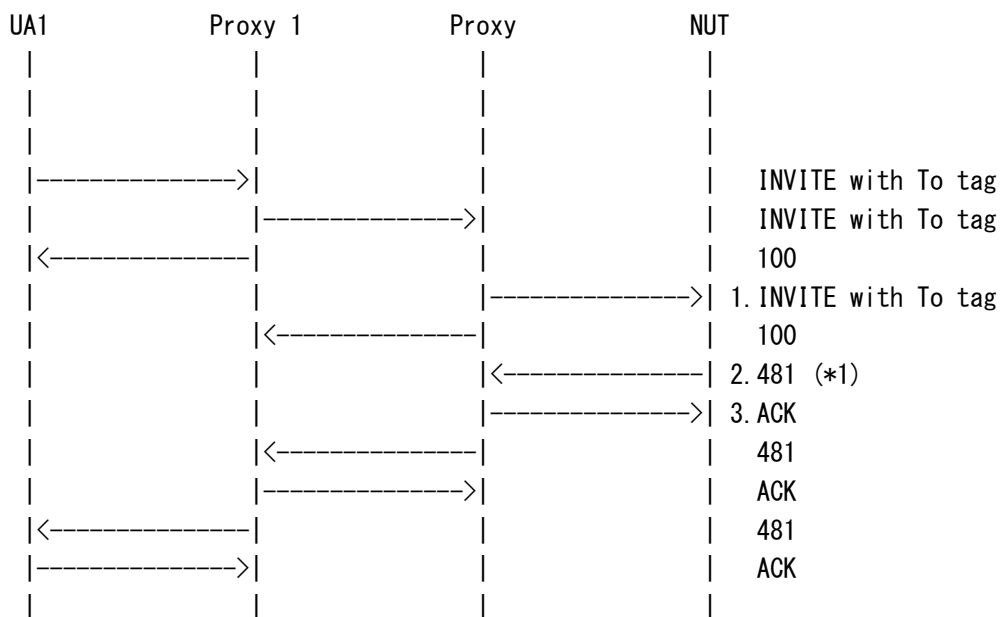
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



- ## [OBSERVABLE RESULTS]

As a SIP Message,  
See `generic_message`

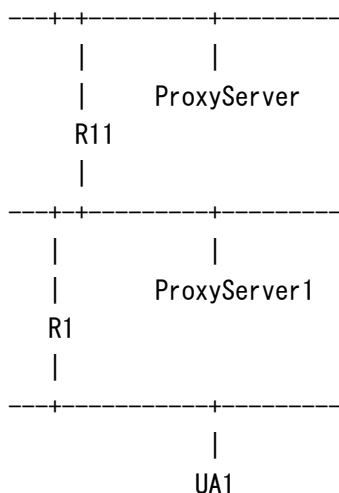
```
- Status-Line:
  See generic_response
  Status-Code: Must be "481". [RFC3261-12-59]
```

**[REFERENCE]**

408



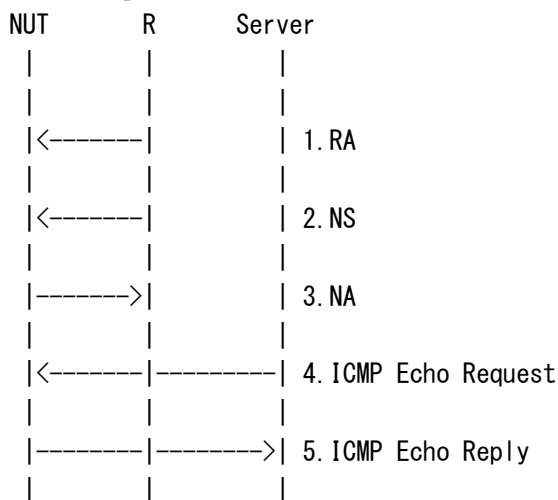




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

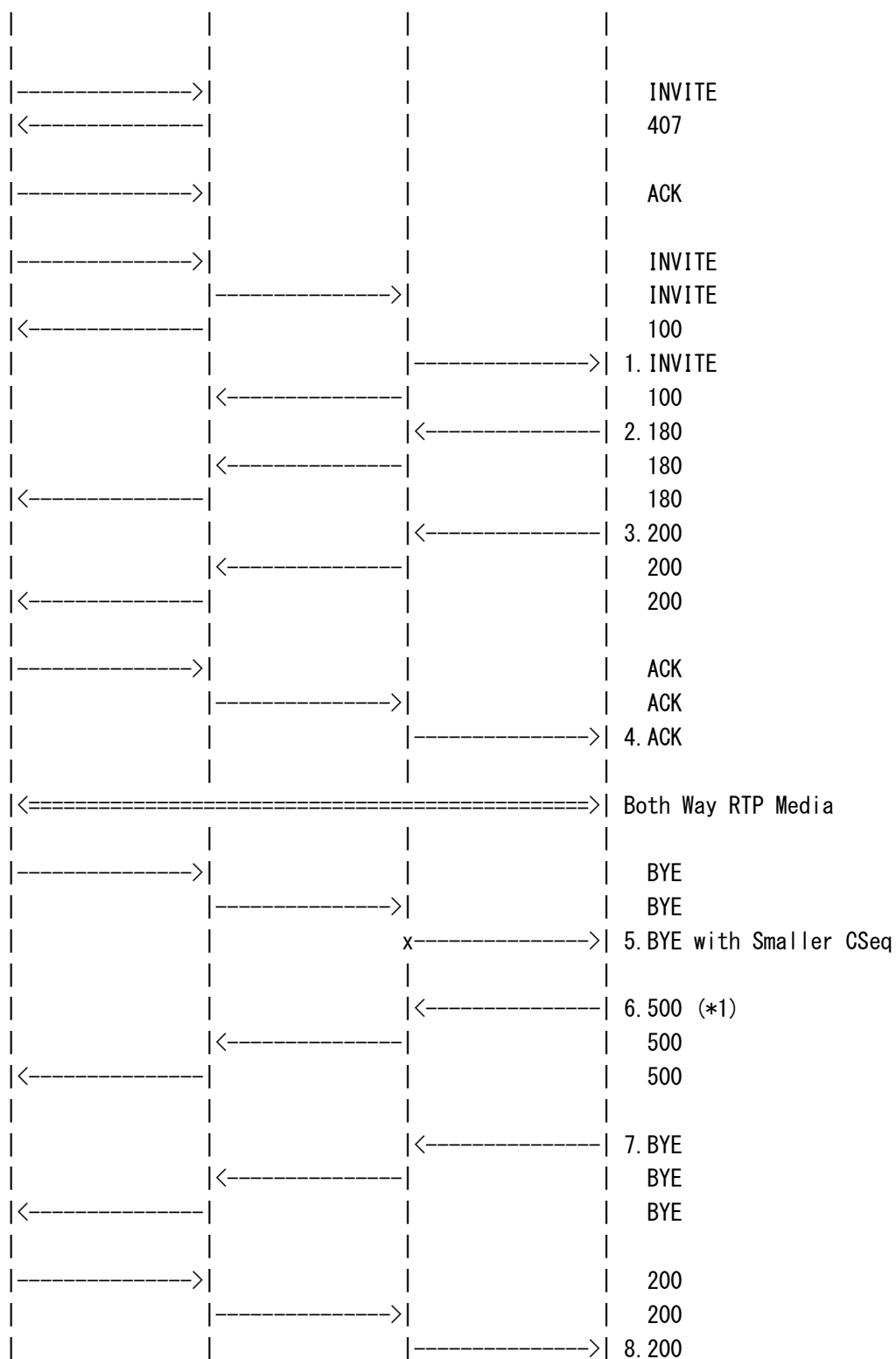
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

UA1                      Proxy 1                      Proxy                      NUT



1. Send INVITE.
2. Receive 180 Ringing.



3. Receive 200 OK.
4. Send ACK.
5. Send BYE.
6. Receive 500 Server Internal Error. (\*1)
7. Receive BYE.
8. Send 200 OK.

5. BYE

\* Cseq number is smaller than 4.ACK's

### **[OBSERVABLE RESULTS]**

\*1:500 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "500". [RFC3261-12-61]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

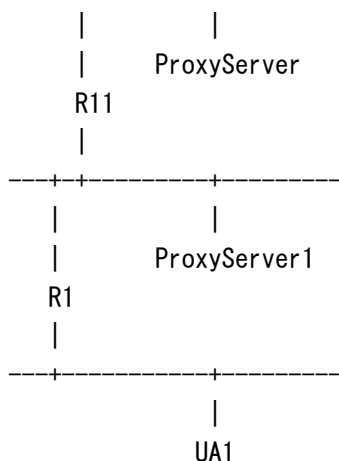
### **[REFERENCE]**

[RFC3261-12-59, 60, 61, 62]

#### **12.2.2 UAS Behavior**

If the remote sequence number is empty, it **MUST** be set to the value of the sequence number in the CSeq header field value in the request. If the remote sequence number was not empty, but the sequence number of the request is lower than the remote sequence number, the request is out of order and **MUST** be rejected with a 500 (Server Internal Error) response. If the remote sequence number was not empty, and the sequence number of the request is greater than the remote sequence number, the request is in order. It is possible for the CSeq sequence number to be higher than the remote sequence number by

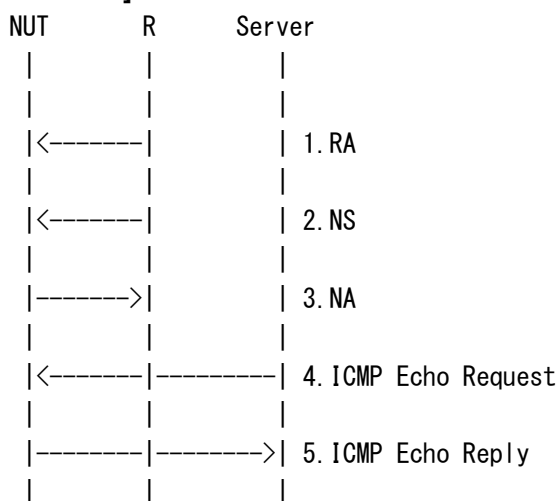




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

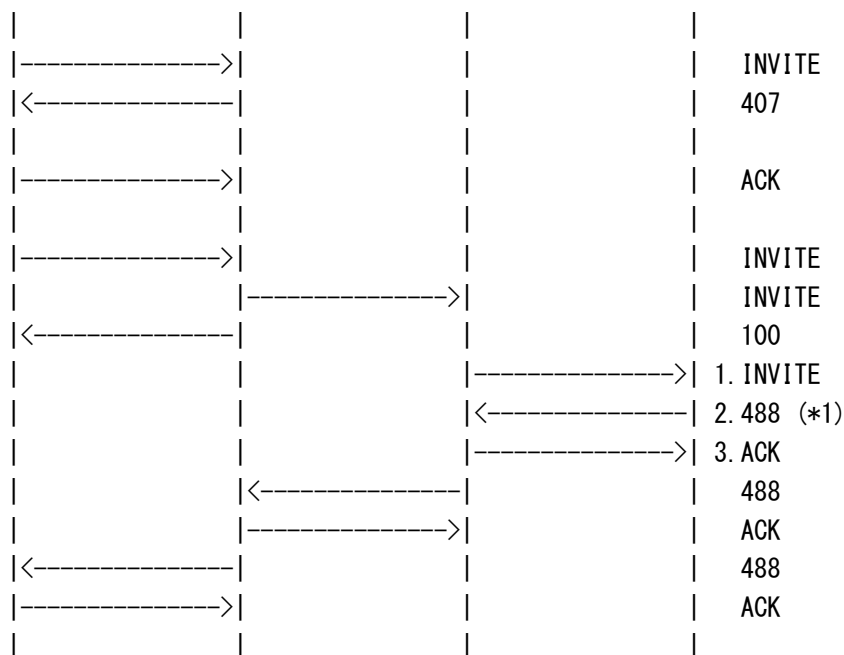
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 488 Not Acceptable Here. (\*1)
3. Send ACK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 151
  
```

v=0



o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 999999999  
a=rtpmap:999999999 PCMU/8000000

#### **[OBSERVABLE RESULTS]**

\*1:488 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "488". [RFC3261-13-34]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Warning  
Should exist. [RFC3261-13-35]

#### **[REFERENCE]**

[RFC3261-13-34, 35]

13.3.1.3 The INVITE is Rejected

A UAS rejecting an offer contained in an INVITE SHOULD return a 488 (Not Acceptable Here) response. Such a response SHOULD include a Warning header field value explaining why the offer was rejected.

### **4.8.7 UA-9-2-7 - INVITE with RFC2543 syntax**

#### **[NAME]**

UA-9-2-7 - INVITE with RFC2543 syntax



### [PURPOSE]

Verify that a UA properly processes an INVITE request with RFC2543 syntax.

### [REQUIREMENT]

Only when a NUT supports the function of backwards compatibility.

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

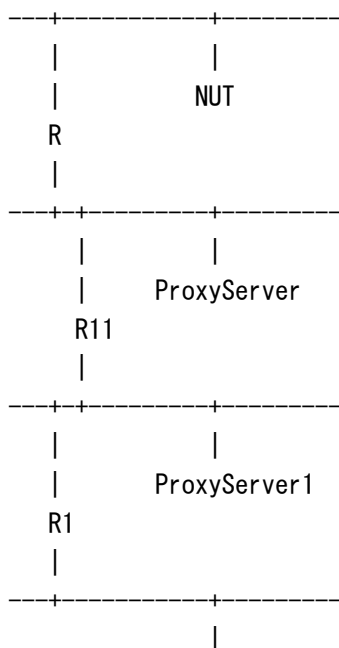
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]



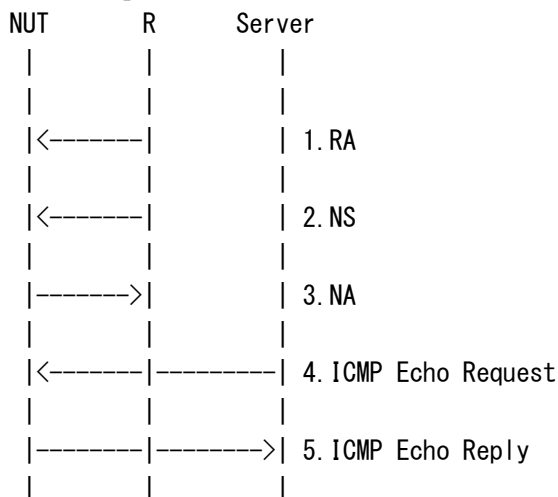


UA1

### [CONFIGURATION for NUT]

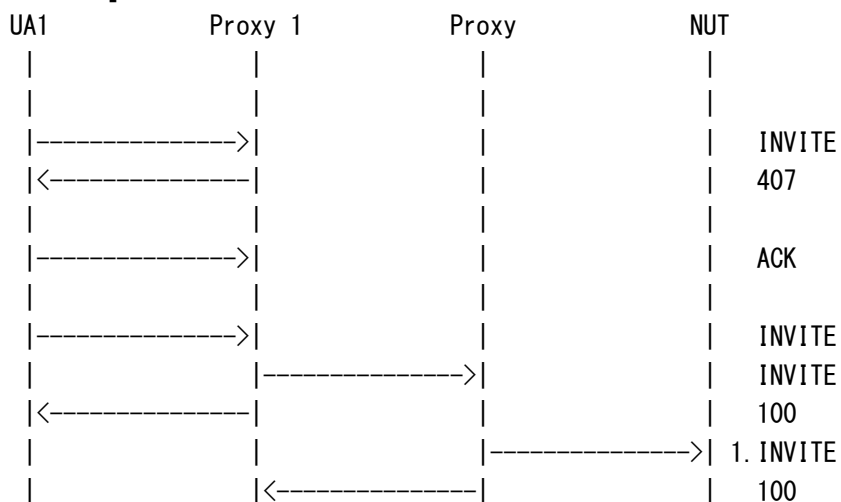
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

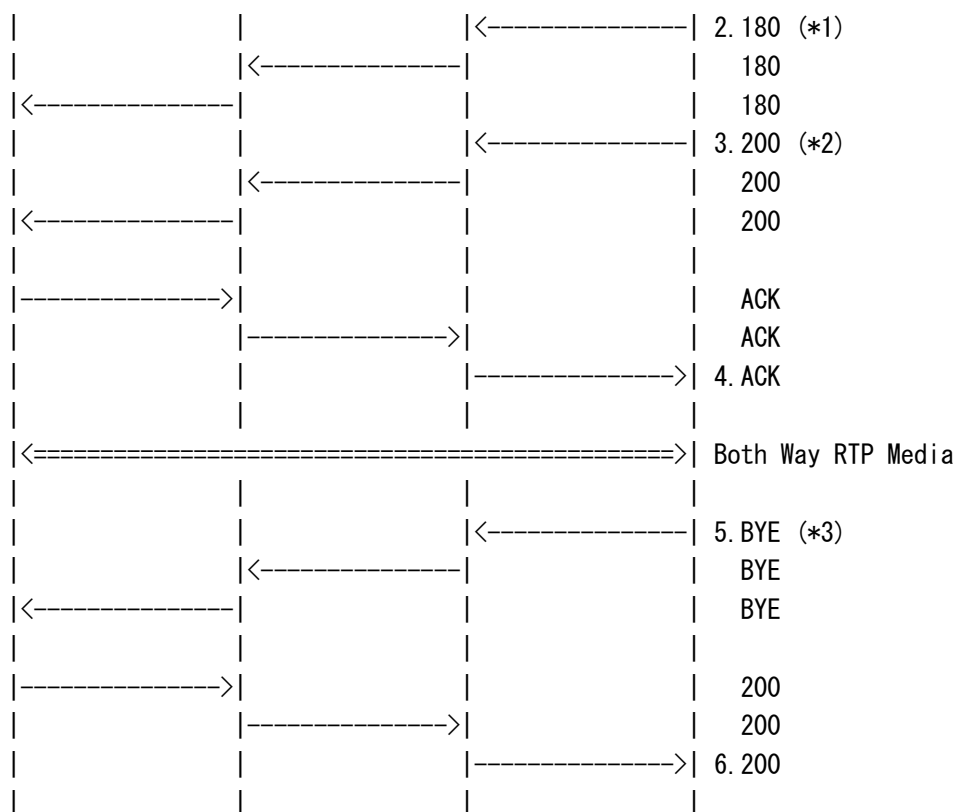
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE. (\*3)
6. Send 200 OK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060
Via: SIP/2.0/UDP ssl.atlanta.example.com:5060
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060
    ;received=3ffe:501:ffff:1::1
Record-Route: <sip:ss.under.test.com:lr>,
    <sip:ssl.atlanta.example.com:lr>
From: UA1 <sip:UA1@atlanta.example.com>
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
    
```



CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp

v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

### **[OBSERVABLE RESULTS]**

\*\* this scenario doesn't check these entities(because sending invalid parameters)  
Via(branch), From, To

\*1:180 response from NUT. [Optional]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]



\*2:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP

\*3:BYE request from NUT.

As a SIP Message,  
See generic\_message



As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request.  
[RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

- \* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

## [REFERENCE]

### 3.4.1 INVITE with RFC2543 syntax

This is a legal message per RFC 2543 (and several bis versions) which should be accepted by RFC 3261 elements which want to maintain backwards compatibility.

- o There is no branch parameter at all on the Via header field value
- o There is no From tag
- o There is no explicit Content-Length (The body is assumed to be all octets in the datagram after the null-line)
- o There is no Max-Forwards header field

## 4.9 Response message

### 4.9.1 UA-10-1-1 - Session Progress response

#### [NAME]

UA-10-1-1 - Session Progress response

**[PURPOSE]**

Verify that a NUT properly processes a Session Progress response.

**[REQUIREMENT]**

NONE

**[TARGET]**

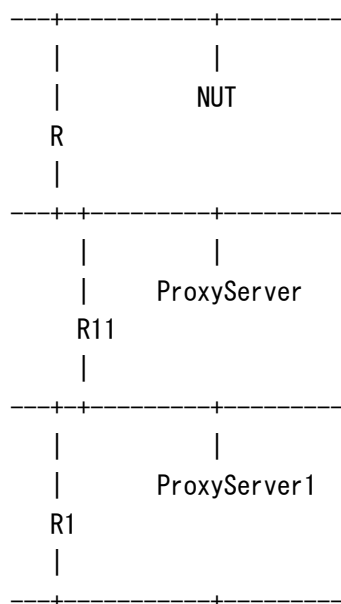
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com:lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

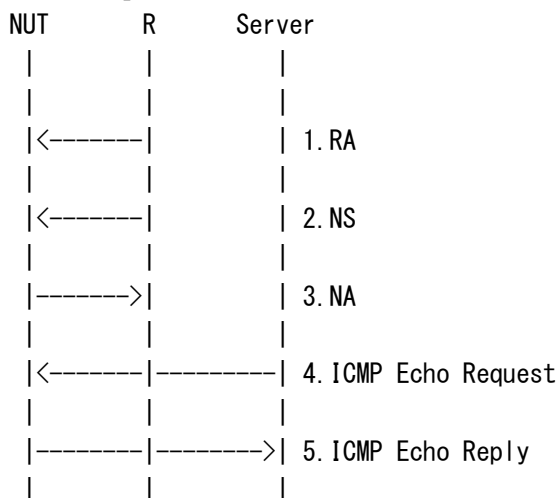
**[TOPOLOGY]**

UA1

### [CONFIGURATION for NUT]

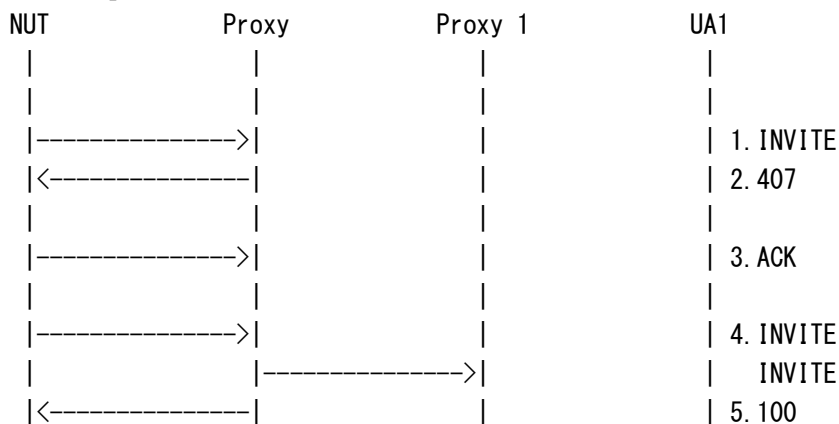
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

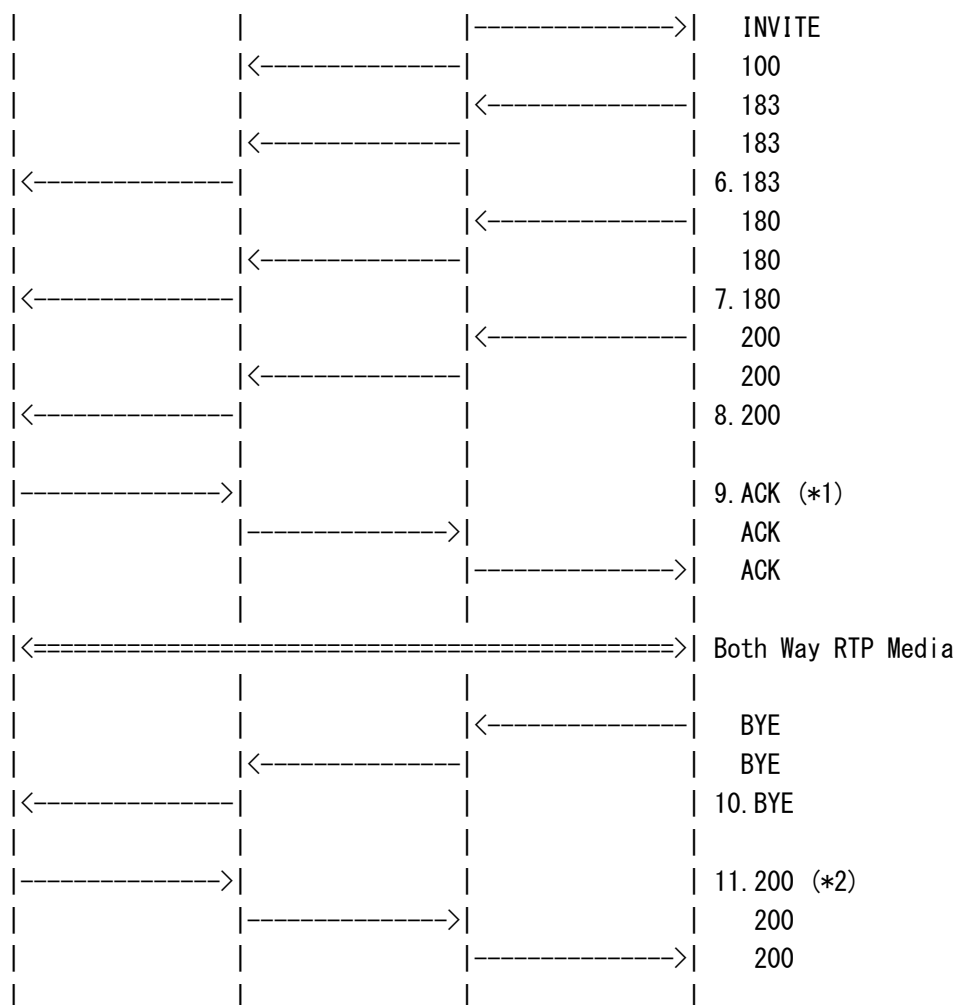
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





- === Message example ===

## SIP/2.0 183 Proceeding





Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
Contact: <sip:UA1@client.atlanta.example.com>  
CSeq: 2 INVITE  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

#### **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

\* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]



\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

## [REFERENCE]

[RFC3261-8-43, 44, 45, 46]

### 8.1.3.2 Unrecognized Responses

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 (Bad Request) response code. A



UAC MUST treat any provisional response different than 100 that it does not recognize as 183 (Session Progress). A UAC MUST be able to process 100 and 183 responses.

#### 4.9.2 UA-10-2-1 - Non-Forwarding of request upon receipt of 503

##### [NAME]

UA-10-2-1 - Not Forwarding of request upon receipt of 503

##### [PURPOSE]

Verify that a NUT properly doesn't forward a request when receiving a 503 response.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

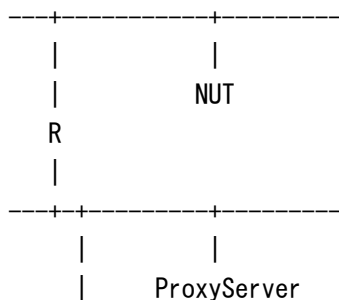
##### [PARAMETER]

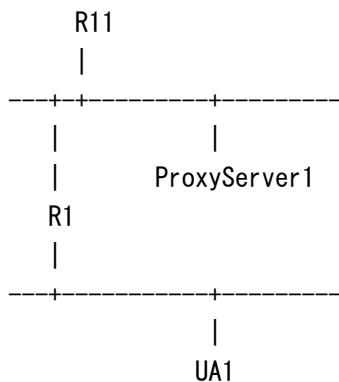
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

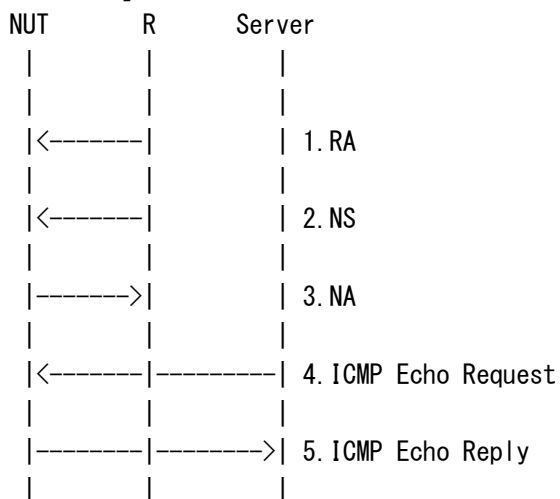




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

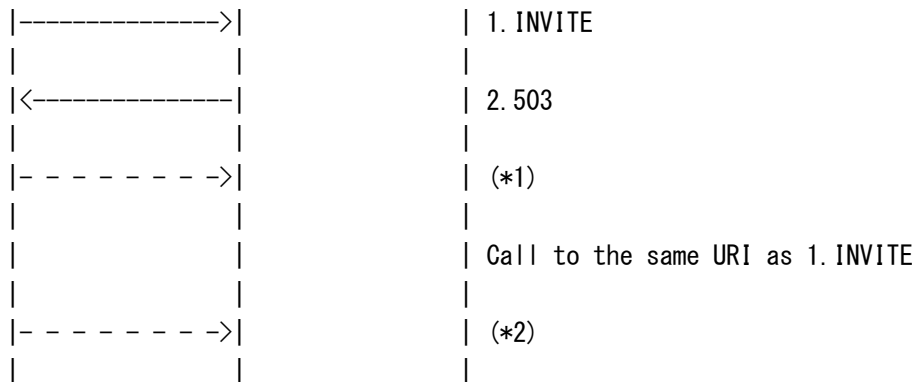
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 503 Service Unavailable.
- (\*1)
- Call to the same URI as 1. INVITE
- (\*2)

**=== Message example ===**

**2. 503 Service Unavailable Proxy -> NUT**

SIP/2.0 503 Service Unavailable  
 Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKbf9f46  
 ;received=3ffe:501:ffff:5::X  
 From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
 To: UA1 <sip:UA1@atlanta.example.com>;tag=53fHlqlQ4  
 Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
 Retry-After: 3600  
 CSeq: 4 INVITE  
 Content-Length: 0

**[OBSERVABLE RESULTS]**

\*\* this scenario checks only timing (message format is not checked)

\*1:after 503 response from Proxy.

Should not send ACK request for "1. INVITE". [RFC3261-21-28]

\*2:after 503 response from Proxy.

Should not send any request, even if User calls to the same URI as



"1.INVITE". [RFC3261-21-28]

#### [REFERENCE]

[RFC3261-21-27, 28]

21.5.4 503 Service Unavailable

A client (proxy or UAC) receiving a 503 (Service Unavailable) SHOULD attempt to forward the request to an alternate server. It SHOULD NOT forward any other requests to that server for the duration specified in the Retry-After header field, if present.

### 4.9.3 UA-10-2-2 - Receipt of SDP answer in a provisional response

#### [NAME]

UA-10-2-2 – Receipt of SDP answer in a provisional response

#### [PURPOSE]

Verify that a NUT properly when receiving a SDP answer in a provisional response.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

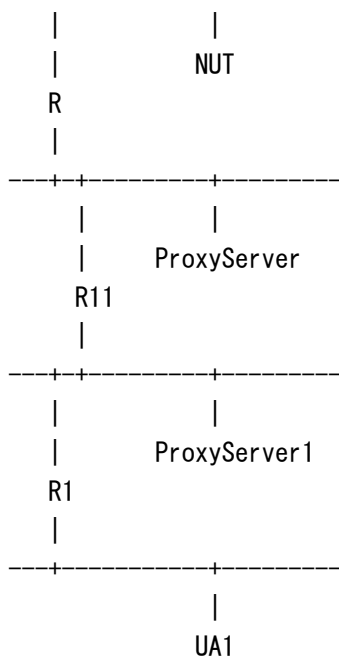
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

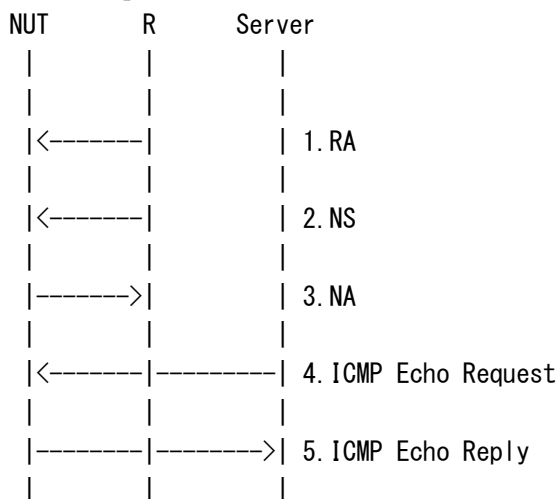
-----+-----+-----



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

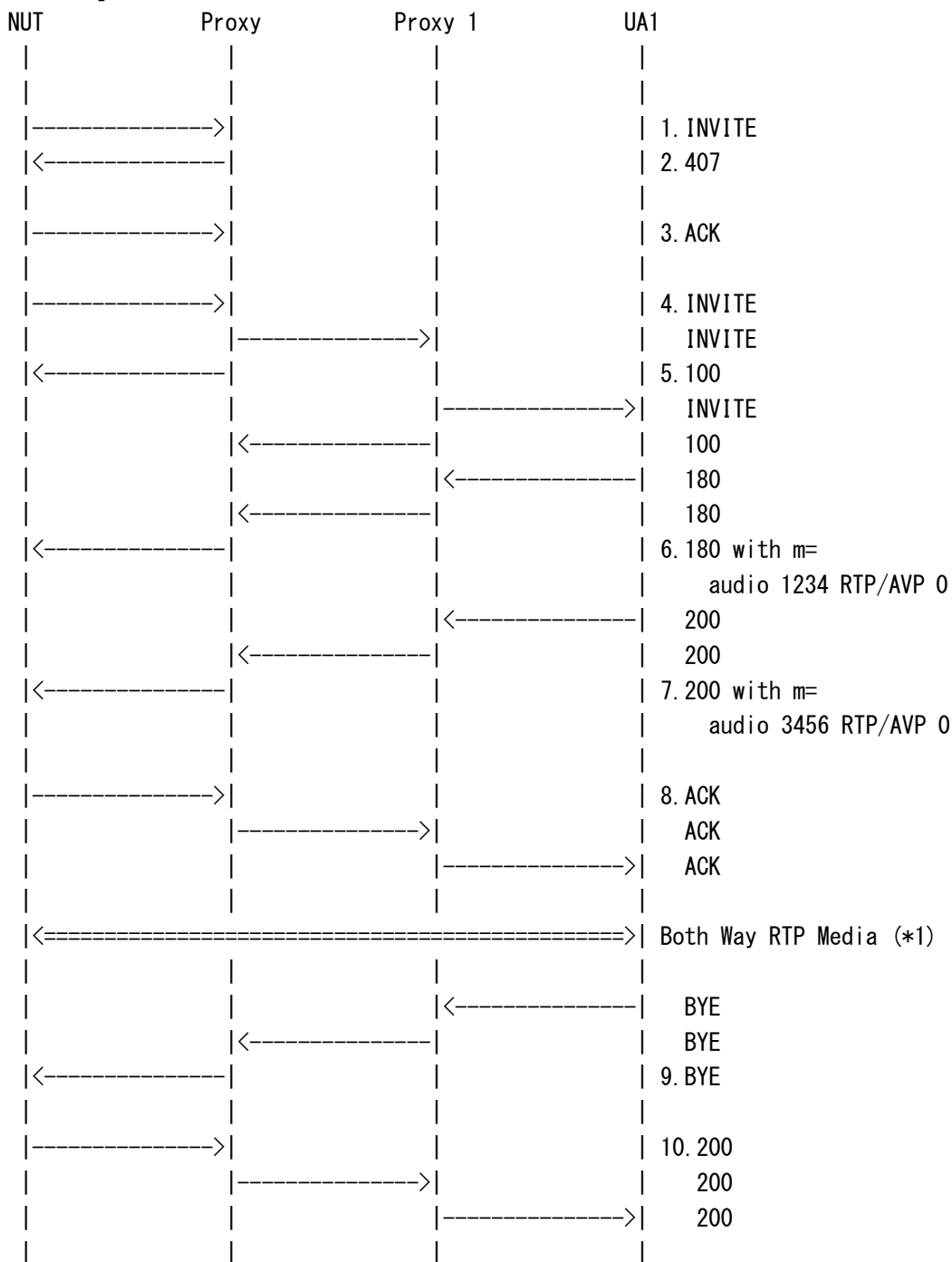
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.

4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407 Proxy Authorization Required.



3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
- (\*1)
9. Send BYE.
10. Receive 200 OK.

**=== Message example ===**

**6. 180 Ringing Proxy -> NUT**

SIP/2.0 180 Ringing  
(snip)  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 1234 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

**7. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
(snip)  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

\* the port number in the m= line is different from 6.180's.



## [OBSERVABLE RESULTS]

\*1:Both Way RTP Media established.

Must send RTP packets on the port specified in the "6.180" response.  
[RFC3261-13-7, 8, 9]

## [REFERENCE]

[RFC3261-13-7, 8, 9]

### 13.2.1 Creating the Initial INVITE

- o If the initial offer is in an INVITE, the answer MUST be in a reliable non-failure message from UAS back to UAC which is correlated to that INVITE. For this specification, that is only the final 2xx response to that INVITE. That same exact answer MAY also be placed in any provisional responses sent prior to the answer. The UAC MUST treat the first session description it receives as the answer, and MUST ignore any session descriptions in subsequent responses to the initial INVITE.

## 4.9.4 UA-10-2-3 - Unrecognized response code (2xx)

### [NAME]

UA-10-2-3 - Unrecognized response code (2xx)

### [PURPOSE]

Verify that a NUT properly processes an unrecognized response code in the 2xx class.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

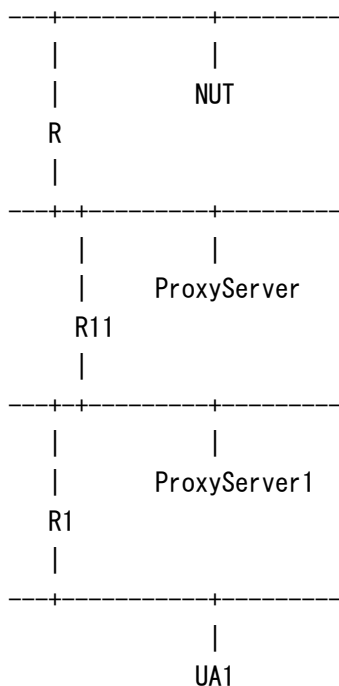
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	ss1.atlanta.example.com;lr

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

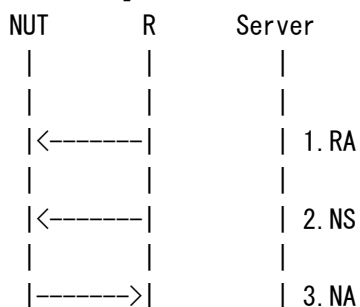
#### [TOPOLOGY]

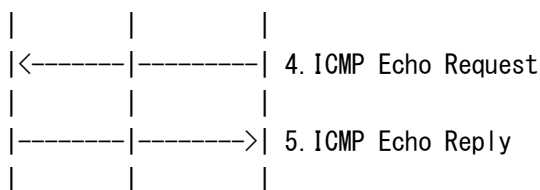


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

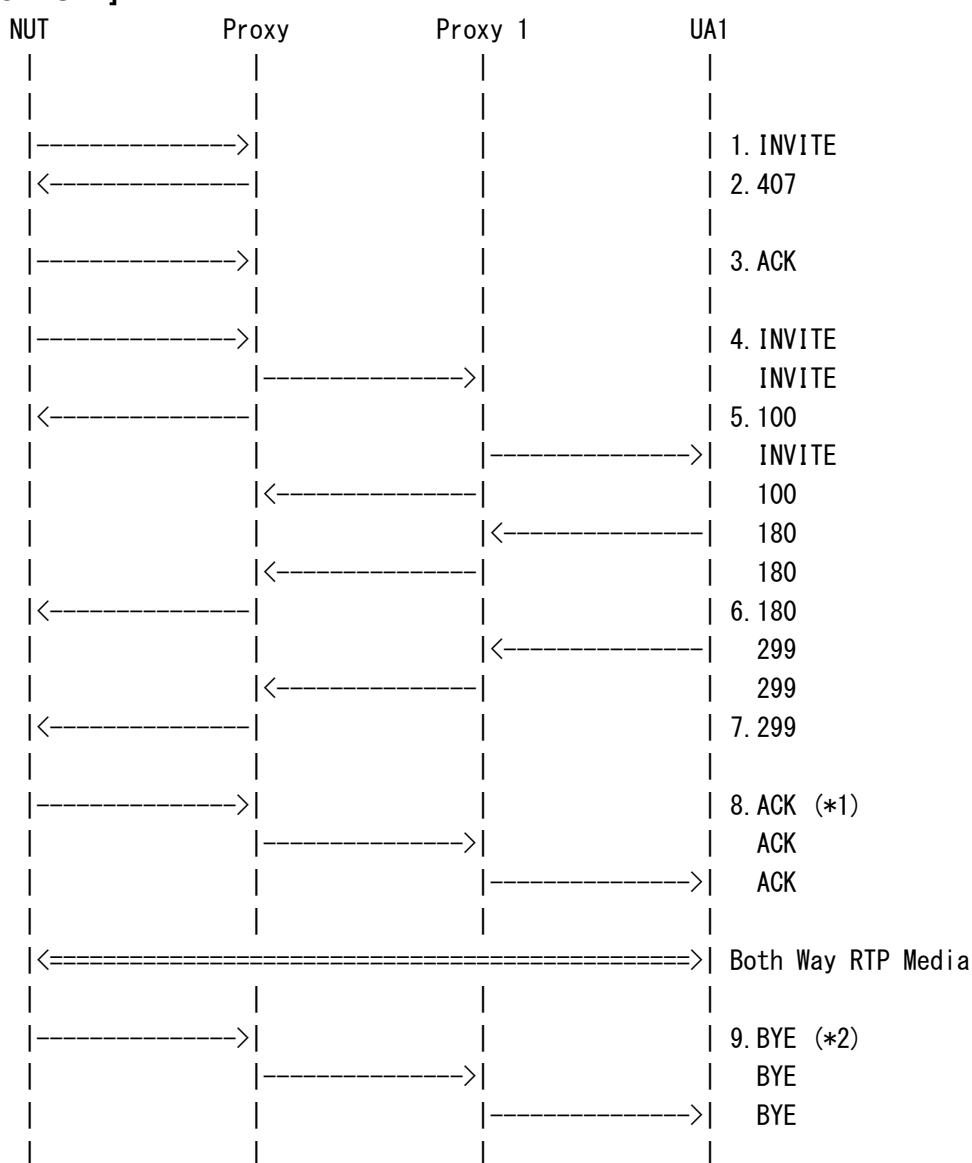
#### [INITIALIZATION]

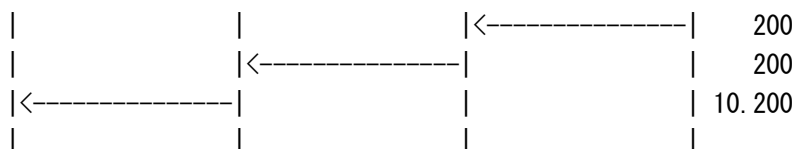




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 299.
8. Receive ACK. (\*1)
9. Receive BYE. (\*2)
10. Send 200 OK.

**=== Message example ===**

**7. 299 Proxy -> NUT**

SIP/2.0 299 OK

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9

;received=3ffe:501:ffff:5::X

Record-Route: <sip:ss1.atlanta.example.com;lr>,

<sip:ss.under.test.com;lr>

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 3848276298220188511@under.test.com

CSeq: 2 INVITE

Contact: <sip:UA1@client.atlanta.example.com>

Content-Type: application/sdp

Content-Length: 147

v=0

o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1

s=-

c=IN IP6 3ffe:501:ffff:1::1

t=0 0

m=audio 3456 RTP/AVP 0

a=rtpmap:0 PCMU/8000

**[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.



Must treat 299 response from Proxy as 200 response. [RFC3261-8-43, 44]

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

\* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]

\* Route  
Must exist. [ORq-2]  
Must contain the Record-Route values of "7.299" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter.  
[RFC3261-12-23, 48]

- Bodies:  
See generic\_ACK  
See generic\_2xx-ACK

\*2:BYE request from NUT.

Must treat 299 response from Proxy as 200 response. [RFC3261-8-43,44]

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:
  - See generic\_request
  - See generic\_BYE
  - Request-URI: Must be the URI of Contact in "7.299" response.  
[RFC3261-12-47]
- Header fields:
  - See generic\_request
- inside of a dialog
  - See generic\_BYE
- \* To
  - tag-param: Must equal that contained in the From header field of "7.299" response. [RFC3261-12-35]
- \* From
  - tag-param: Must equal that contained in the To header field of "1.INVITE".  
[RFC3261-12-37]
- \* Route
  - Must include a Route header field. [RFC3261-12-48]
  - route-param: Must contain the route set values in order, including all parameters. [RFC3261-12-48]
- Bodies:
  - See generic\_BYE

## [REFERENCE]

[RFC3261-8-43, 44, 45, 46]

### 8.1.3.2 Unrecognized Responses

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 (Bad Request) response code. A UAC MUST treat any provisional response different than 100 that it does not recognize as 183 (Session Progress). A UAC MUST be able to process 100 and 183 responses.



## 4.9.5 UA-10-2-4 - Unrecognized response code (4xx)

### [NAME]

UA-10-2-4 - Unrecognized response code (4xx)

### [PURPOSE]

Verify that a NUT properly processes an unrecognized response code in the 4xx class.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

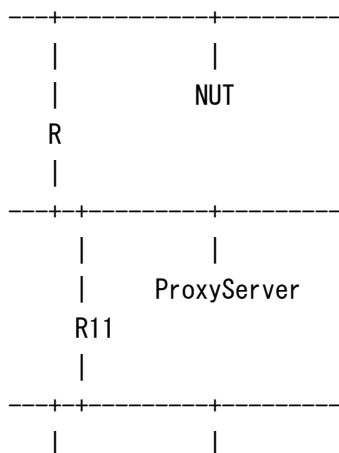
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

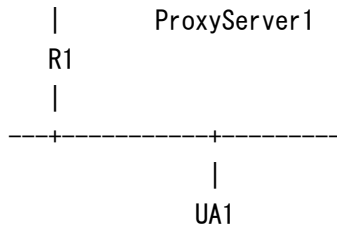
### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]



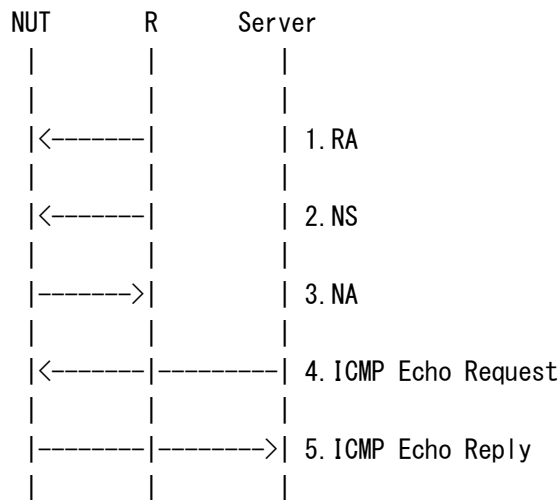




#### [CONFIGURATION for NUT]

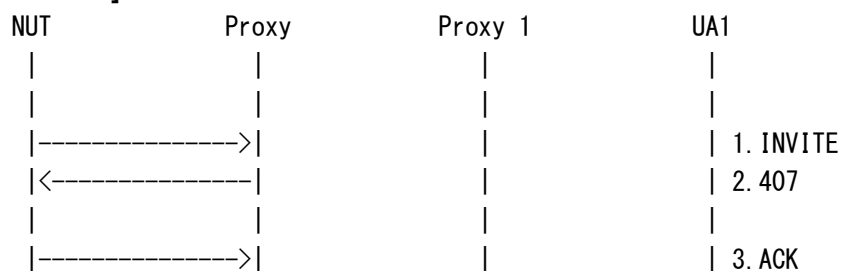
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

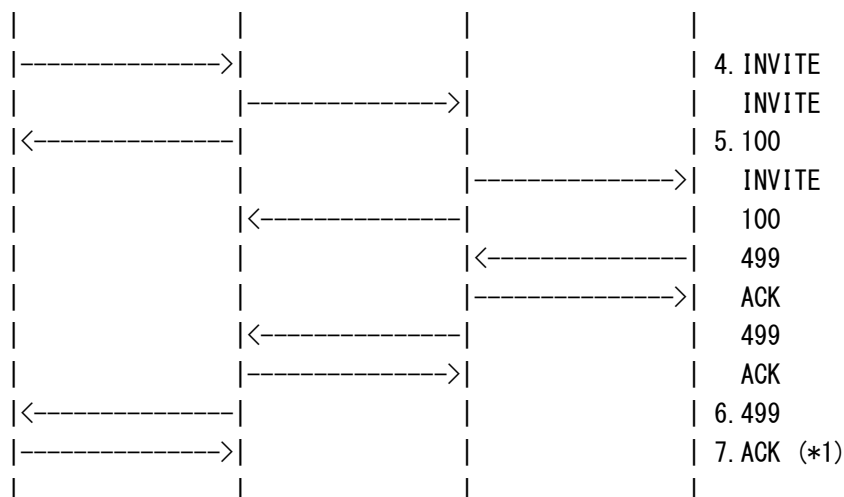
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





- === Message example ===

## SIP/2.0 499 Error

### [OBSERVABLE RESULTS]

Must treat 499 response from Proxy as 400 response. [RFC3261-8-43,44]

As a SIP request,



- Request-Line:
  - See generic\_request
  - See generic\_ACK
  - See generic\_non2xx-ACK
- Header fields:
  - See generic\_request
- outside of a dialog
  - See generic\_ACK
  - See generic\_non2xx-ACK
- Bodies:
  - See generic\_ACK
  - See generic\_non2xx-ACK

## **[REFERENCE]**

[RFC3261-8-43, 44, 45, 46]

### **8.1.3.2 Unrecognized Responses**

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 (Bad Request) response code. A UAC MUST treat any provisional response different than 100 that it does not recognize as 183 (Session Progress). A UAC MUST be able to process 100 and 183 responses.

## **4.9.6 UA-10-2-5 - Unrecognized response code (5xx)**

### **[NAME]**

UA-10-2-5 - Unrecognized response code (5xx)

### **[PURPOSE]**

Verify that a NUT properly processes an unrecognized response code in the 5xx class.

### **[REQUIREMENT]**

NONE

### **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

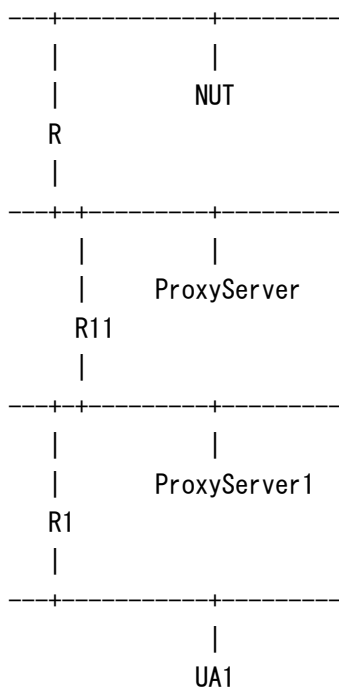
# **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

# **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# **[TOPOLOGY]**

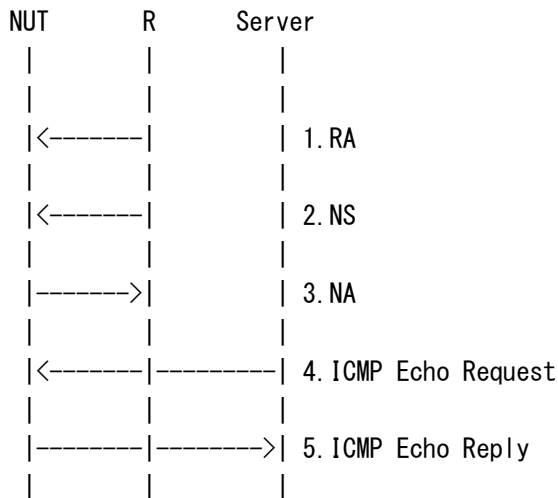


# **[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com

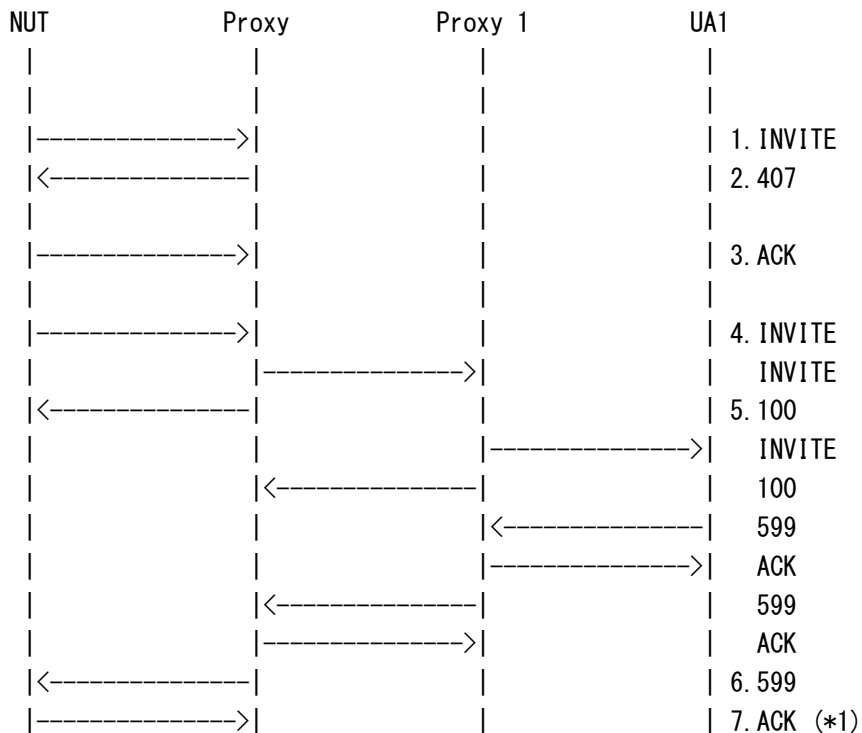
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)
-------------	--------------------------------

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 599.
7. Receive ACK. (\*1)

**=== Message example ===**

**6. 599 Proxy -> NUT**

SIP/2.0 599 Error

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43

;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf

Call-ID: 3848276298220188511@under.test.com

CSeq: 1 INVITE

Content-Length: 0

**[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

Must treat 599 response from Proxy as 500 response. [RFC3261-8-43,44]

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_non2xx-ACK

- Header fields:

See generic\_request

- outside of a dialog

See generic\_ACK

See generic\_non2xx-ACK



- Bodies:
  - See generic\_ACK
  - See generic\_non2xx-ACK

#### [REFERENCE]

[RFC3261-8-43, 44, 45, 46]

#### 8.1.3.2 Unrecognized Responses

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 (Bad Request) response code. A UAC MUST treat any provisional response different than 100 that it does not recognize as 183 (Session Progress). A UAC MUST be able to process 100 and 183 responses.

### 4.9.7 UA-10-2-6 - Unrecognized response code (6xx)

#### [NAME]

UA-10-2-6 - Unrecognized response code (6xx)

#### [PURPOSE]

Verify that a NUT properly processes an unrecognized response code in the 6xx class.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

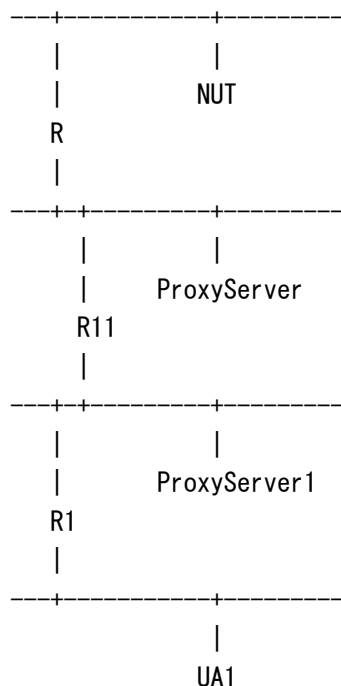
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
-----------	---

R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

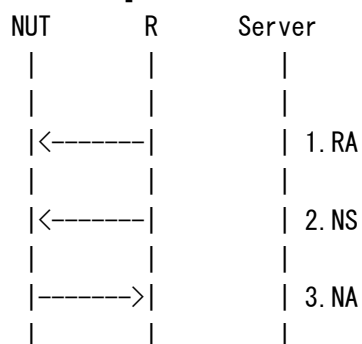
#### [TOPOLOGY]



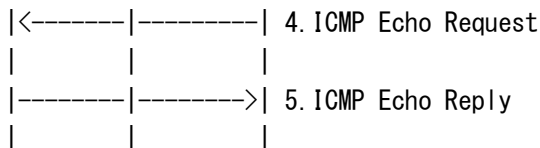
#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

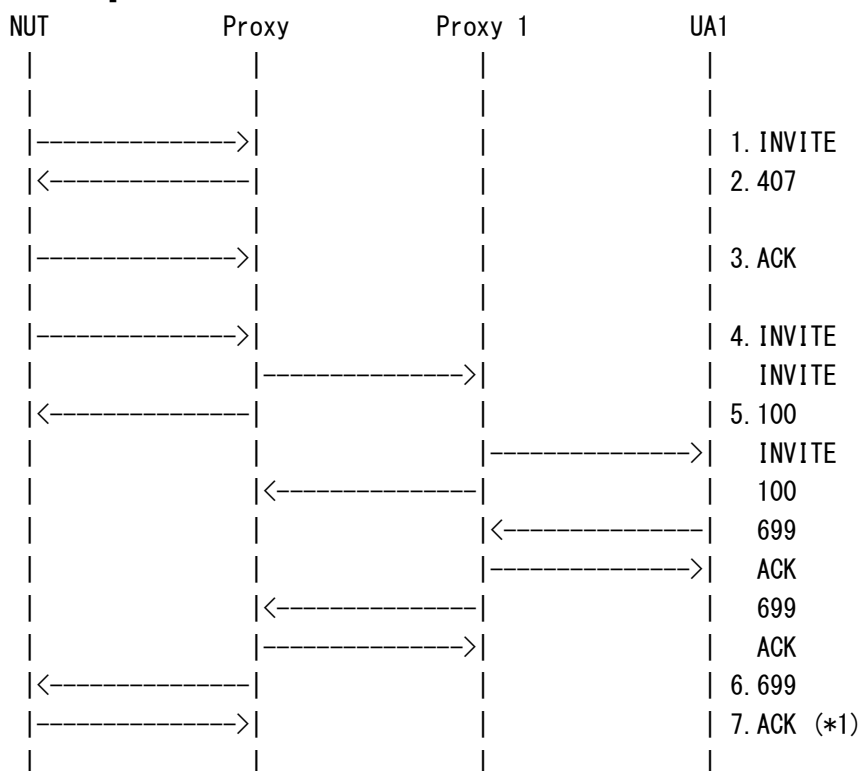






1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 699.
7. Receive ACK. (\*1)

=== Message example ===

6. 699 Proxy -> NUT



SIP/2.0 699 Error

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b43

;received=3ffe:501:ffff:5::X

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=3flal12sf

Call-ID: 3848276298220188511@under.test.com

CSeq: 1 INVITE

Content-Length: 0

## **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

Must treat 699 response from Proxy as 600 response. [RFC3261-8-43,44]

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

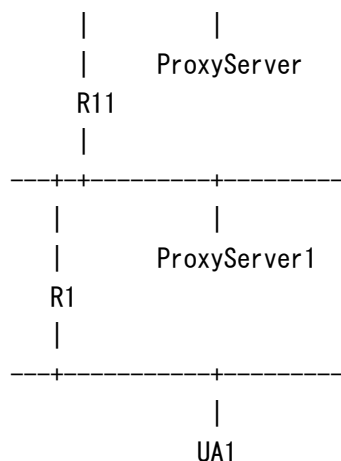
## **[REFERENCE]**

[RFC3261-8-43, 44, 45, 46]

### **8.1.3.2 Unrecognized Responses**

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely

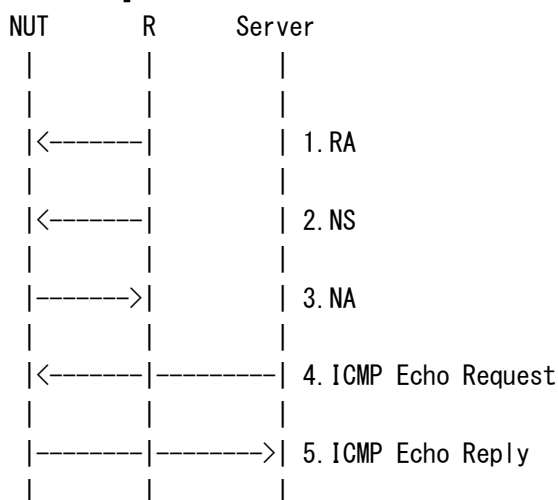




### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

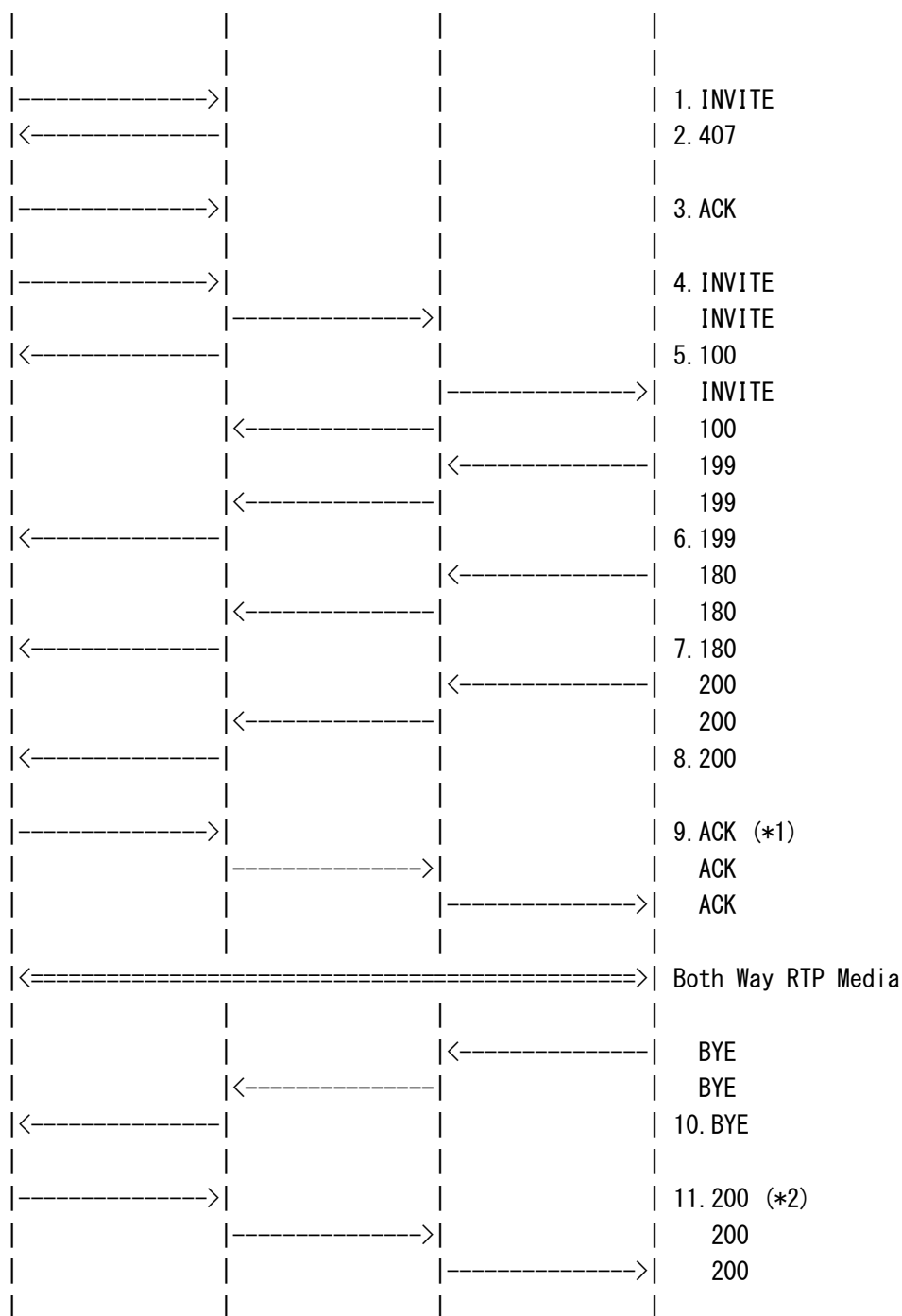
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]

NUT                      Proxy                      Proxy 1                      UA1



1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 199.



7. Send 180 Ringing.
8. Send 200 OK.
9. Receive ACK. (\*1)
10. Send BYE.
11. Receive 200 OK. (\*2)

**=== Message example ===**

**6. 199 Proxy -> NUT**

SIP/2.0 199 Going on  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
Contact: <sip:UA1@client.atlanta.example.com>  
CSeq: 2 INVITE  
Content-Length: 0

**[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

\* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]



Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261.22.2.1]

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

## [REFERENCE]

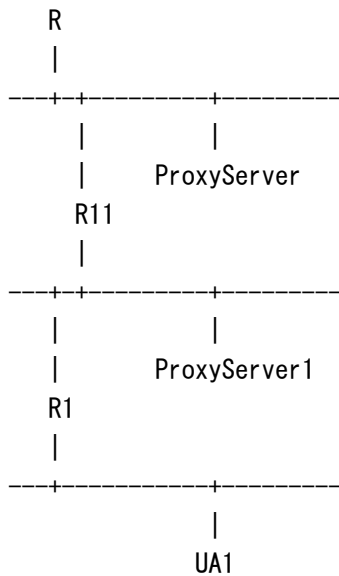
[RFC3261-8-43, 44, 45, 46]

### 8.1.3.2 Unrecognized Responses

A UAC MUST treat any final response it does not recognize as being equivalent to the x00 response code of that class, and MUST be able to process the x00 response code for all classes. For example, if a UAC receives an unrecognized response code of 431, it can safely



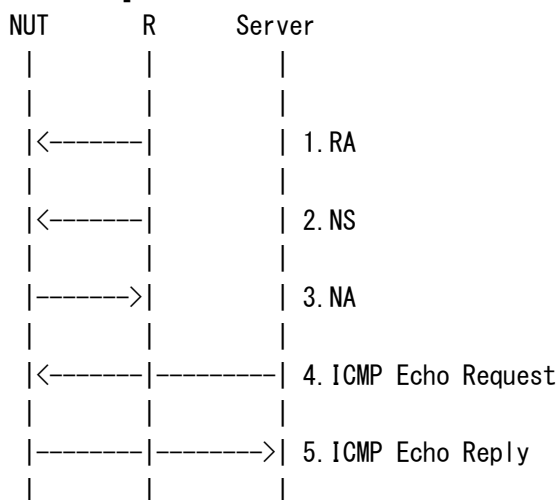




#### [CONFIGURATION for NUT]

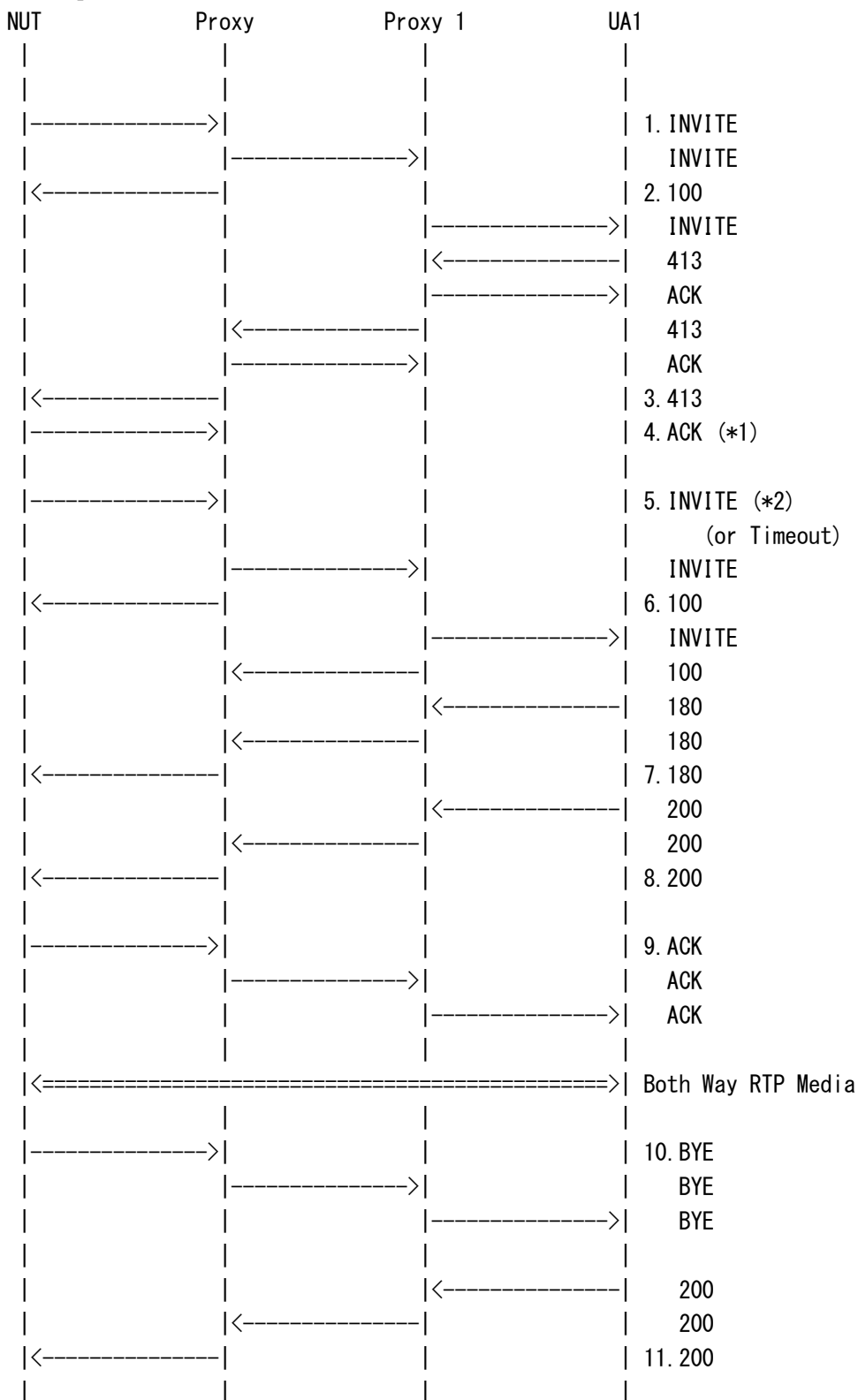
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]



1. Receive INVITE.
2. Send 100 Trying.
3. Send 413 Request Entity Too Large.
4. Receive ACK. (\*1)
5. Receive INVITE. (\*2)
6. Send 100 Trying.
7. Send 180 Ringing.
8. Send 200 OK.
9. Receive ACK.
10. Receive BYE.
11. Send 200 OK.

#### **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

Must send ACK for 413 response.

As a SIP Message,  
See generic\_message

As a SIP request,  
- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK

- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

\*2: INVITE request from NUT.

The length of body message Must be less than that of previous INVITE.  
[RFC3261-8-45]

## [REFERENCE]

[RFC3261-8-58]

### 8.1.3.5 Processing 4xx Responses

If a 413 (Request Entity Too Large) response is received (Section 21.4.11), the request contained a body that was longer than the UAS was willing to accept. If possible, the UAC SHOULD retry the request, either omitting the body or using one of a smaller length.

## 4.9.10 UA-10-2-9 - Processing request without credentials after receipt of 403 (Forbidden) response

### [NAME]

UA-10-2-9 - 403 (Forbidden) response

### [PURPOSE]

Verify that a NUT properly processes a request without credential after receiving a 403 (Forbidden) response.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

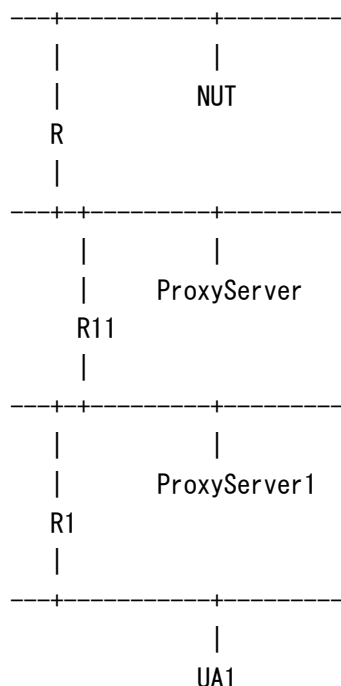
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

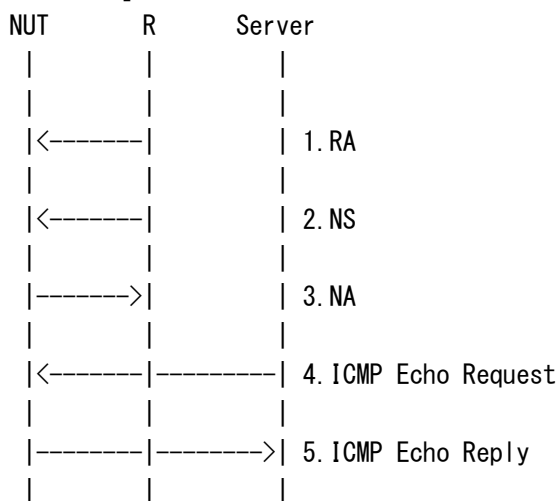
### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

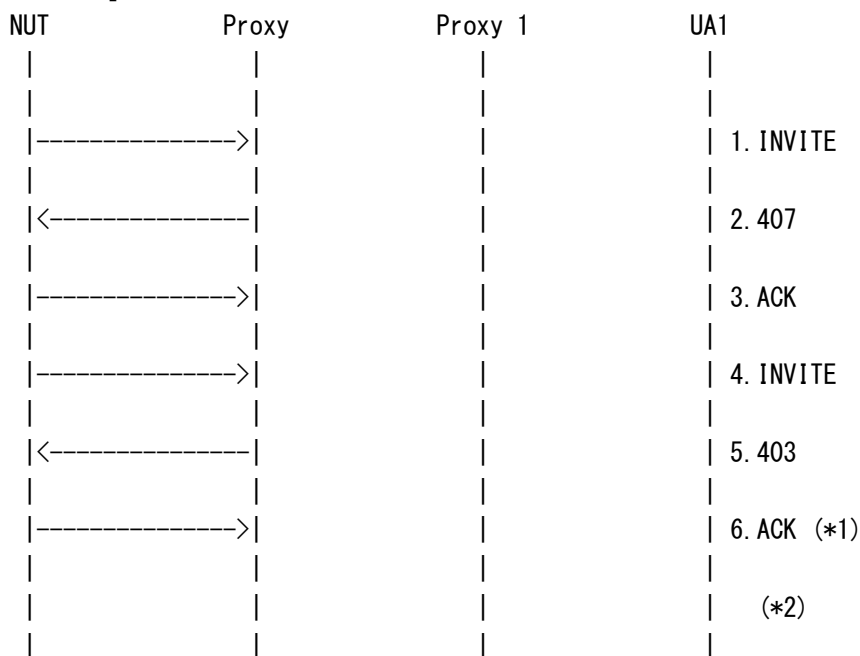
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.

3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 403 Forbidden.
6. Receive ACK. (\*1)
- (\*2)

#### [OBSERVABLE RESULTS]

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK



- Header fields:  
See generic\_request
- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK
- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

\*2:after ACK request from NUT.

Should not send INVITE. [RFC3261-21-11]  
Must not re-attempt requests with the credentials  
that have just been rejected (though the request may be retried if  
the nonce was stale). [RFC3261-22-14]

#### [REFERENCE]

[RFC3261-21-11]  
21.4.4 403 Forbidden

The server understood the request, but is refusing to fulfill it.  
Authorization will not help, and the request SHOULD NOT be repeated.

[RFC3261-22-14]  
22.1 Framework

Finally, note that even if a UAC can locate credentials that are associated with the proper realm, the potential exists that these credentials may no longer be valid or that the challenging server will not accept these credentials for whatever reason (especially when "anonymous" with no password is submitted). In this instance a server may repeat its challenge, or it may respond with a 403 Forbidden. A UAC MUST NOT re-attempt requests with the credentials that have just been rejected (though the request may be retried if the nonce was stale).

#### 4.9.11 UA-10-2-10 - Unsupported Require header field

[NAME]



## UA-10-2-10 - Unsupported Require header field

### [PURPOSE]

Verify that a NUT properly processes an unsupported extension in a Require header field.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

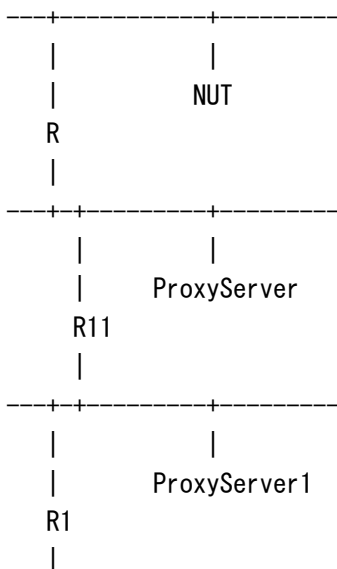
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

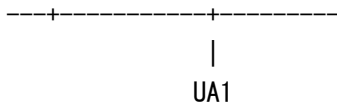
### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]



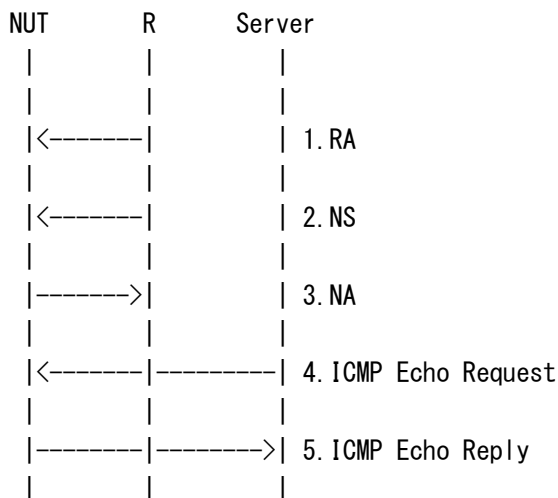




#### [CONFIGURATION for NUT]

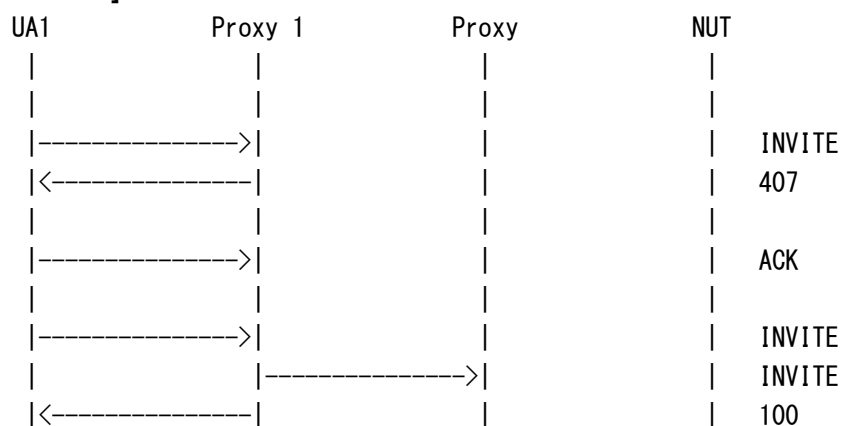
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

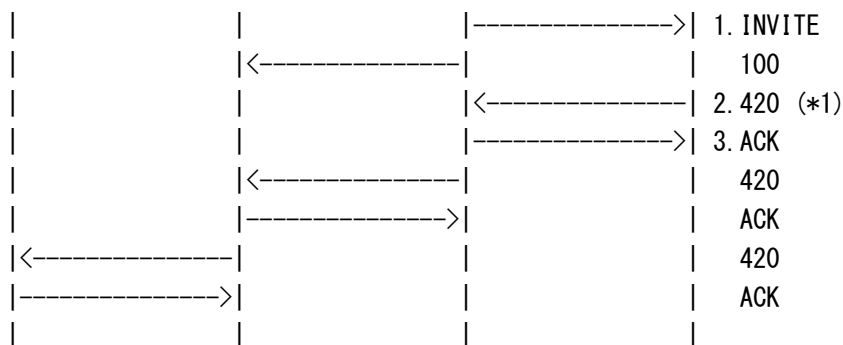
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 420 Bad Extension. (\*1)
3. Send ACK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Require: 999rel
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 151

v=0
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
  
```



## [OBSERVABLE RESULTS]

\*1:420 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "420". [RFC3261-8-78, RFC3261-21-17]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Unsupported:  
Must exist. [RFC3261-8-79, RFC3261-21-12]  
option-tag: Must include "999rel". [RFC3261-21-17]

## [REFERENCE]

[RFC3261-21-17]

21.4.15 420 Bad Extension

The server did not understand the protocol extension specified in a Proxy-Require (Section 20.29) or Require (Section 20.32) header field field. The server MUST include a list of the unsupported extensions in an Unsupported header field in the response. UAC processing of this response is described in Section 8.1.3.5.

## 4.10 Dialog

### 4.10.1 UA-11-1-1 - CANCEL for unmatched requests

#### [NAME]

UA-11-1-1 - CANCEL for unmatched requests

#### [PURPOSE]

Verify that a NUT properly processes a CANCEL request because not finding a matching

transaction for the CANCEL.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

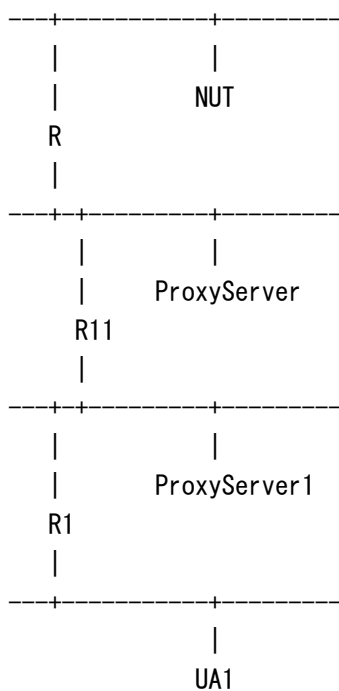
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

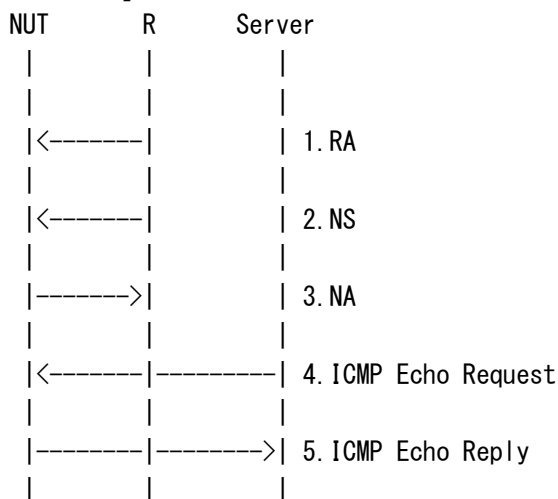
#### [TOPOLOGY]



# [CONFIGURATION for NUT]

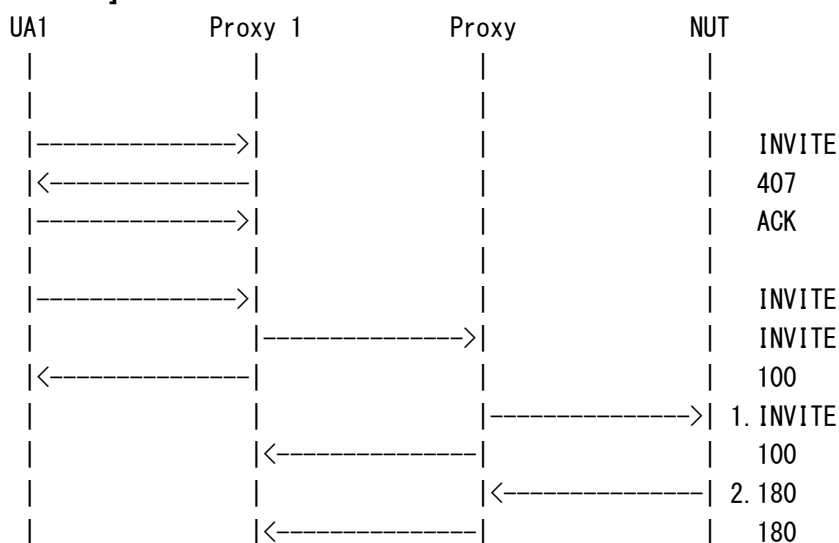
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

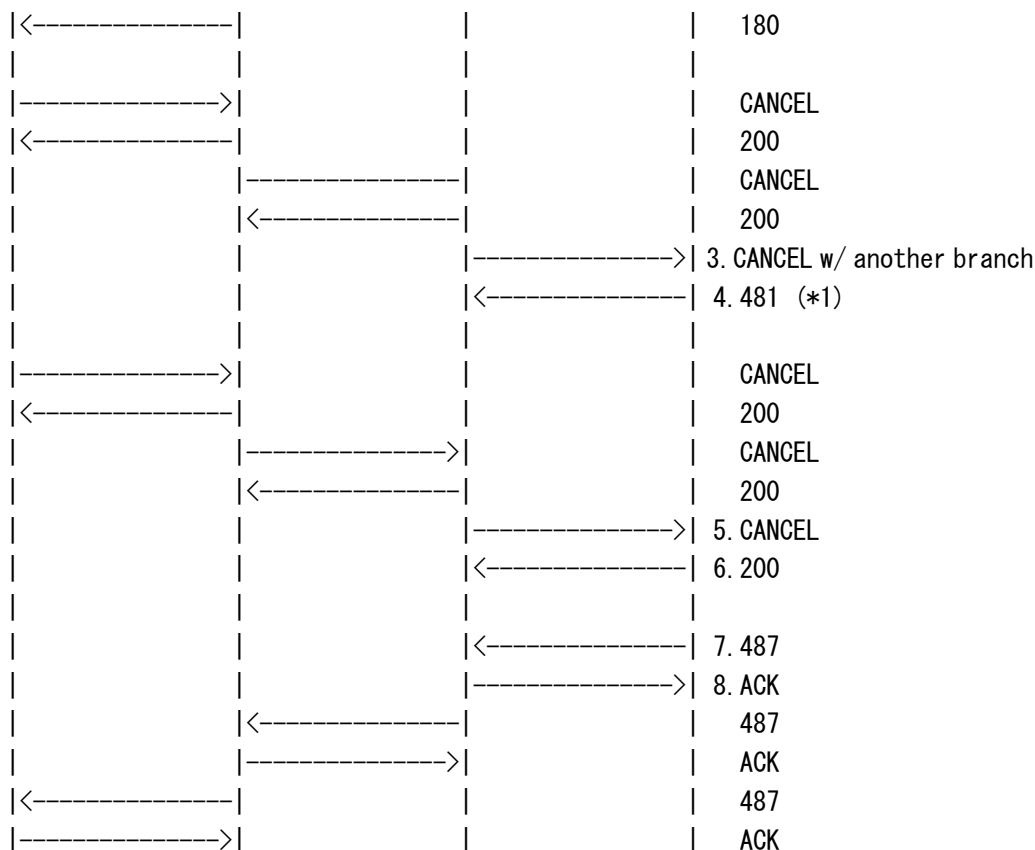
# [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Send CANCEL.
4. Receive 481 Call Leg/Transaction Does Not Exist. (\*1)
5. Send CANCEL.
6. Receive 200 OK.
7. Receive 487 Request Terminated.
8. Send ACK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss2.node.under.test.com:5060;branch=z9hG4bK721e418c4.1
(snip)

```

##### 3. CANCEL Proxy -> NUT

```

CANCEL sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.9

```



(snip)

\* Only the branch ID in the top Via field is different from 1.INVITE's.

#### **[OBSERVABLE RESULTS]**

\*1:481 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "481". [RFC3261-9-14]
- Header fields:  
See generic\_response
  - \* Via
    - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
    - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

#### **[REFERENCE]**

[RFC3261-9-14, 15]

##### **9.2 Server Behavior**

If the UAS did not find a matching transaction for the CANCEL according to the procedure above, it SHOULD respond to the CANCEL with a 481 (Call Leg/Transaction Does Not Exist). If the transaction for the original request still exists, the behavior of the UAS on receiving a CANCEL request depends on whether it has already sent a final response for the original request. If it has, the CANCEL request has no effect on the processing of the original request, no effect on any session state, and no effect on the responses generated for the original request. If the UAS has not issued a final response for the original request, its behavior depends on the method of the original request. If the original request was an INVITE, the UAS SHOULD immediately respond to the INVITE with a 487 (Request Terminated). A CANCEL request has no impact on the processing of transactions with any other method defined in this specification.

## 4.10.2 UA-11-1-2 - Request without a tag in a From header field

### [NAME]

UA-11-1-2 - Request without a tag in a From header field

### [PURPOSE]

Verify that a NUT properly processes a request that doesn't contain a tag in a From header field.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

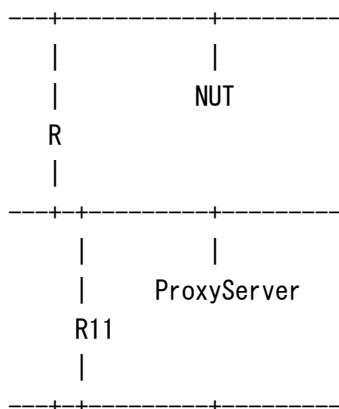
### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

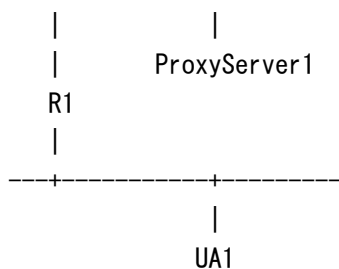
### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

### [TOPOLOGY]



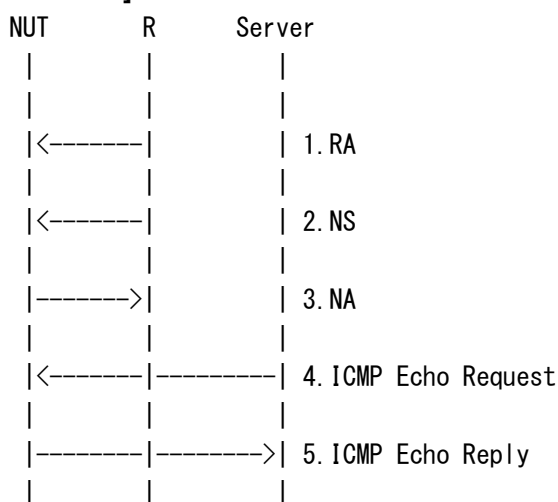




#### [CONFIGURATION for NUT]

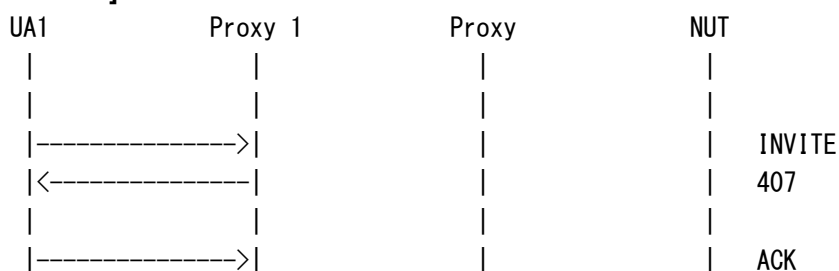
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

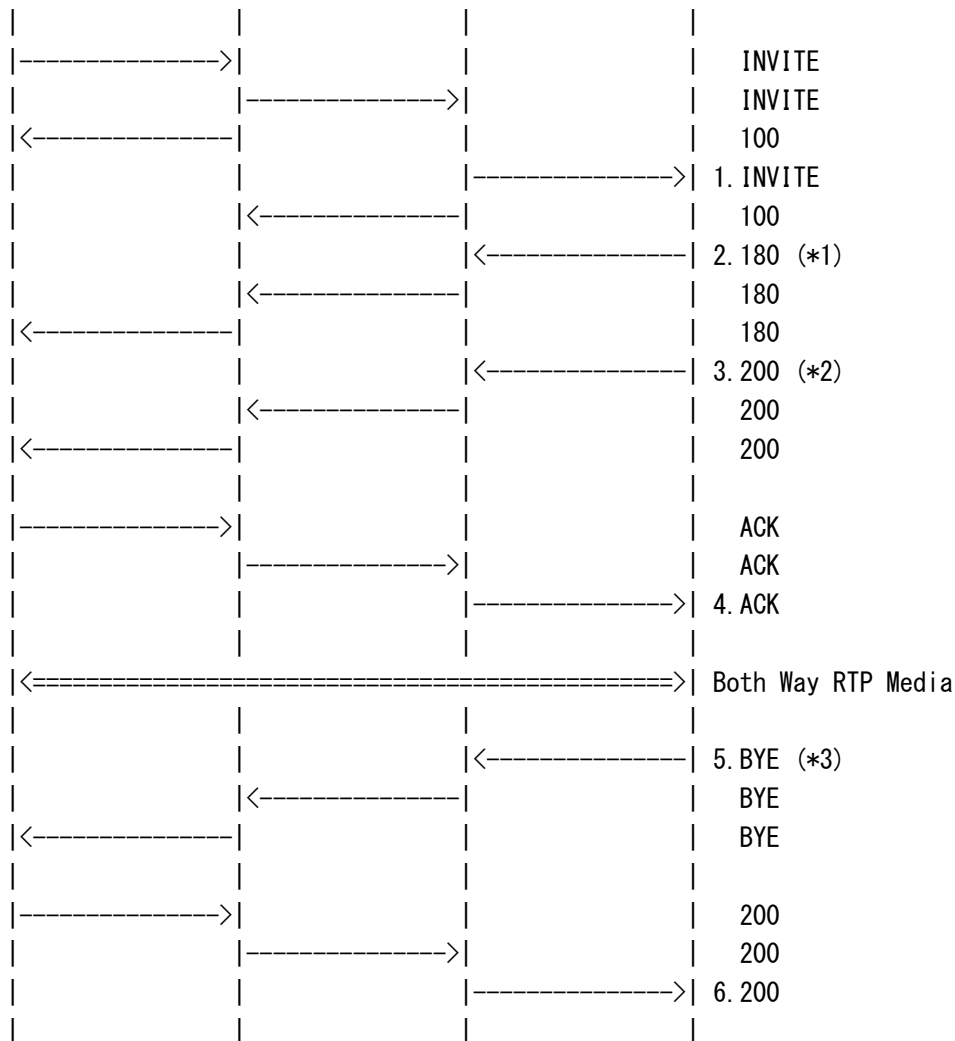
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE. (\*3)
6. Send 200 OK.

#### === Message example ===

##### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
(snip)
From: UA1 <sip:UA1@atlanta.example.com>
To: NUT <sip:NUT@under.test.com>
(snip)
  
```



\* No From tag.

### **2.180 Ringing NUT -> Proxy**

SIP/2.0 180 Ringing

(snip)

From: UA1 <sip:UA1@atlanta.example.com>

To: NUT <sip:NUT@under.test.com>;tag=314159

(snip)

\* No From tag.

### **3.200 OK NUT -> Proxy**

SIP/2.0 200 OK

(snip)

From: UA1 <sip:UA1@atlanta.example.com>

To: NUT <sip:NUT@under.test.com>;tag=314159

(snip)

\* No From tag.

### **4.ACK Proxy -> NUT**

ACK sip:NUT@node.under.test.com SIP/2.0

(snip)

From: UA1 <sip:UA1@atlanta.example.com>

To: NUT <sip:NUT@under.test.com>;tag=314159

(snip)

\* No From tag.

### **5.BYE NUT -> Proxy**

BYE sip:UA1@client.atlanta.example.com SIP/2.0

(snip)

From: NUT <sip:NUT@under.test.com>;tag=314159

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.



## 6.200 OK Proxy -> NUT

SIP/2.0 200 OK

(snip)

From: NUT <sip:NUT@under.test.com>;tag=314159

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.

### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:

See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* From

tag-param: Must be omitted from the From header fields. [RFC3261-12-17, RFC3261-12-38]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]



\*2:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* From  
tag-param: Must be omitted from the From header fields. [RFC3261-12-17, RFC3261-12-38]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP

\*3:BYE request from NUT.



As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request.  
[RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must be omitted from the To header fields. [RFC3261-12-17,  
RFC3261-12-38]

\* From

tag-param: Must equal that contained in the To header field of "3.200"  
response. [RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

## [REFERENCE]

[RFC3261-12-12, 13, 14, 15, 16, 17]

12.1.1 UAS behavior

The remote sequence number **MUST** be set to the value of the sequence number in the CSeq header field of the request. The local sequence number **MUST** be empty. The call identifier component of the dialog ID **MUST** be set to the value of the Call-ID in the request. The local tag component of the dialog ID **MUST** be set to the tag in the To field in the response to the request (which always includes a tag), and the remote tag component of the dialog ID **MUST** be set to the tag from the



From field in the request. A UAS MUST be prepared to receive a request without a tag in the From field, in which case the tag is considered to have a value of null.

[RFC3261-12-34, 35, 36, 37, 38]

#### 12.2.1.1 Generating the Request

The URI in the To field of the request MUST be set to the remote URI from the dialog state. The tag in the To header field of the request MUST be set to the remote tag of the dialog ID. The From URI of the request MUST be set to the local URI from the dialog state. The tag in the From header field of the request MUST be set to the local tag of the dialog ID. If the value of the remote or local tags is null, the tag parameter MUST be omitted from the To or From header fields, respectively.

### 4.10.3 UA-11-1-3 - Response without a tag in a To header field

#### [NAME]

UA-11-1-3 - Response without a tag in a To header field

#### [PURPOSE]

Verify that a NUT properly processes a response that doesn't contain a tag in a To header field.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

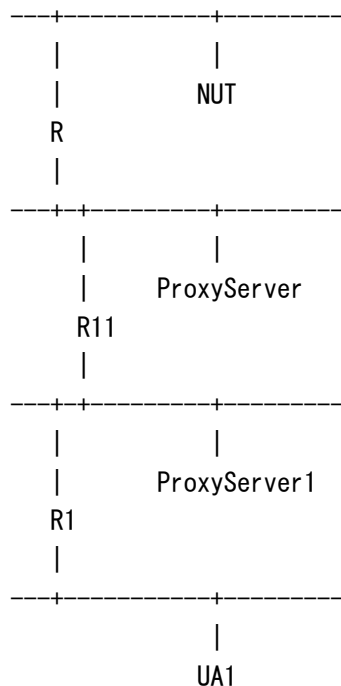
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64

ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

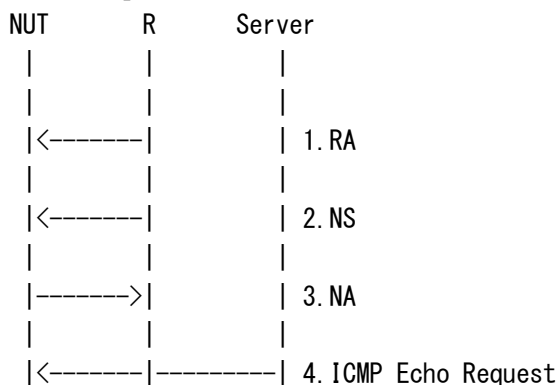
#### [TOPOLOGY]



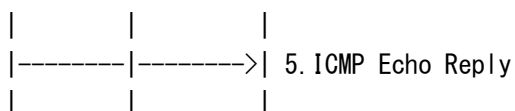
#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

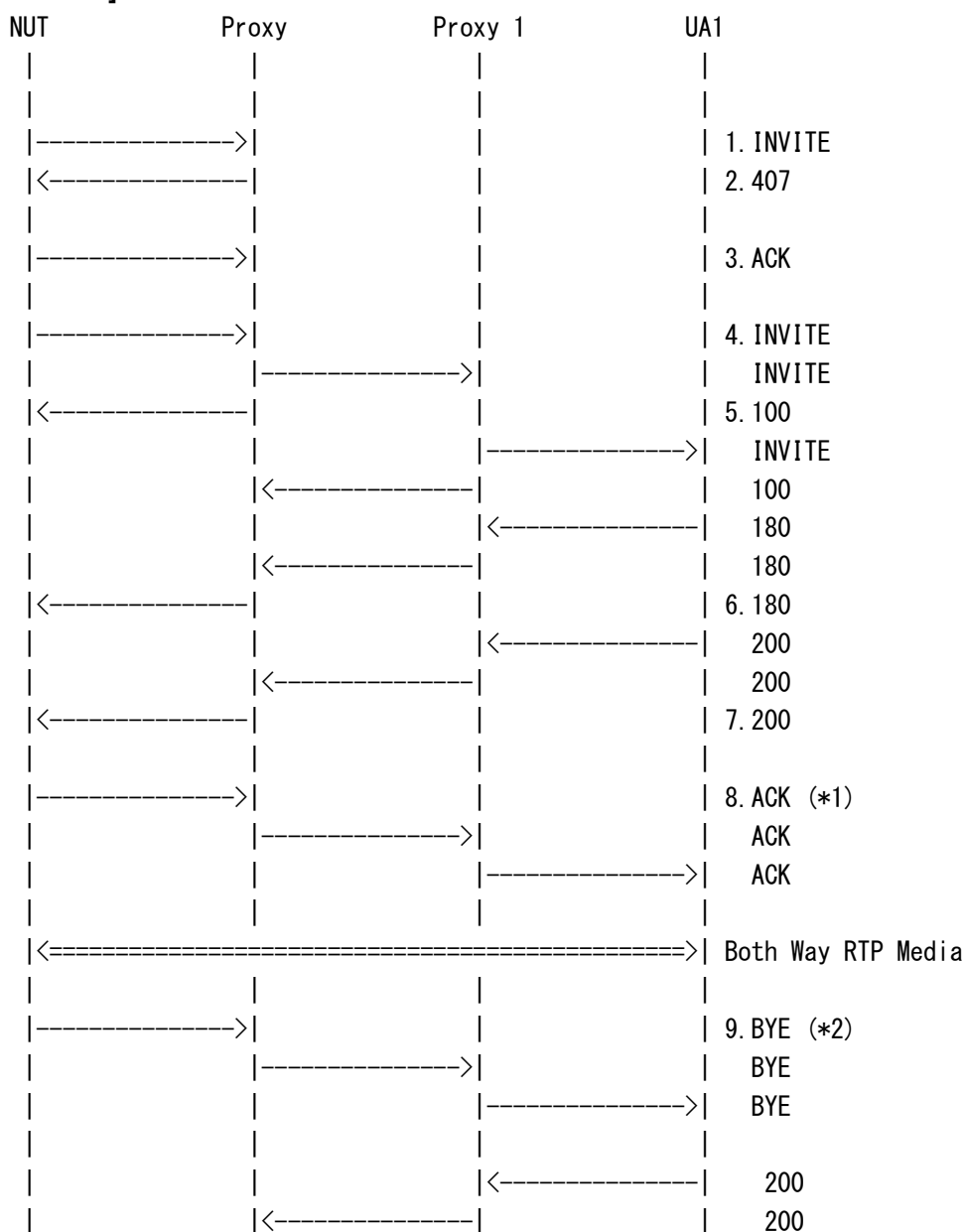






1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





<-----		10. 200

1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK. (\*1)
9. Receive BYE. (\*2)
10. Send 200 OK.

**=== Message example ===**

**6. 180 Ringing Proxy -> NUT**

SIP/2.0 180 Ringing

(snip)

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.

**7. 200 OK Proxy -> NUT**

SIP/2.0 200 OK

(snip)

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.

**8. ACK NUT -> Proxy**

ACK sip:UA1@client.atlanta.example.com SIP/2.0

(snip)

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

(snip)



\* No To tag.

### **9.BYE NUT -> Proxy**

BYE sip:UA1@client.atlanta.example.com SIP/2.0

(snip)

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.

### **10.200 OK Proxy -> NUT**

SIP/2.0 200 OK

(snip)

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>

(snip)

\* No To tag.

## **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

\* Proxy-Authorization



Must exist. [RFC3261.22.3],[RFC3261-22-22]

Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

\* To

tag-param: Must be omitted from the To header fields. [RFC3261-12-31, RFC3261-12-39]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*3:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "7.200" response. [RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must be omitted from the To header fields. [RFC3261-12-31, RFC3261-12-39]

\* From

tag-param: Must equal that contained in the From header field of "1.INVITE" request. [RFC3261-12-37]



\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

## [REFERENCE]

[RFC3261-12-26, 27, 28, 29, 30]

### 12.1.2 UAC Behavior

The local sequence number **MUST** be set to the value of the sequence number in the CSeq header field of the request. The remote sequence number **MUST** be empty (it is established when the remote UA sends a request within the dialog). The call identifier component of the dialog ID **MUST** be set to the value of the Call-ID in the request. The local tag component of the dialog ID **MUST** be set to the tag in the From field in the request, and the remote tag component of the dialog ID **MUST** be set to the tag in the To field of the response. A UAC **MUST** be prepared to receive a response without a tag in the To field, in which case the tag is considered to have a value of null.

[RFC3261-12-34, 35, 36, 37, 38]

### 12.2.1.1 Generating the Request

The URI in the To field of the request **MUST** be set to the remote URI from the dialog state. The tag in the To header field of the request **MUST** be set to the remote tag of the dialog ID. The From URI of the request **MUST** be set to the local URI from the dialog state. The tag in the From header field of the request **MUST** be set to the local tag of the dialog ID. If the value of the remote or local tags is null, the tag parameter **MUST** be omitted from the To or From header fields, respectively.

## 4.10.4 UA-11-1-4 - Multiple 2xx responses

### [NAME]

UA-11-1-4 - Multiple 2xx responses

### [PURPOSE]



Verify that a NUT properly establishes each session when receiving multiple 2xx responses.

**[REQUIREMENT]**

Only when a NUT supports multiple response.

**[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

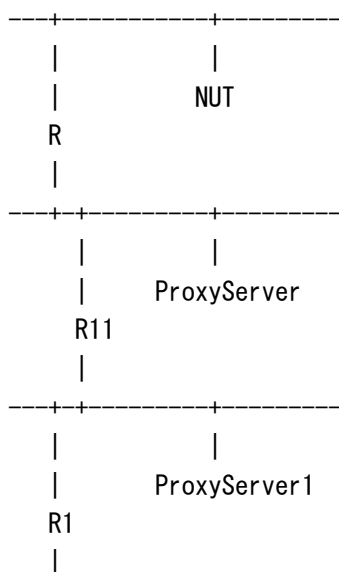
**[PARAMETER]**

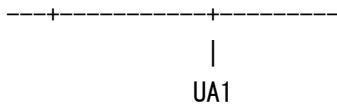
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA11(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
UA11(Contact)	sip:UA11@client1.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

**[TOPOLOGY]**

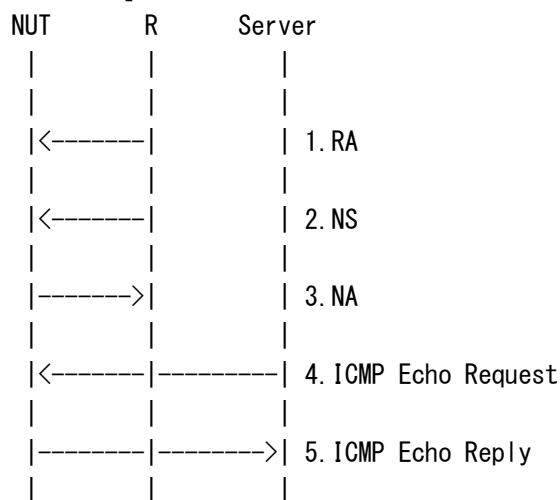




#### [CONFIGURATION for NUT]

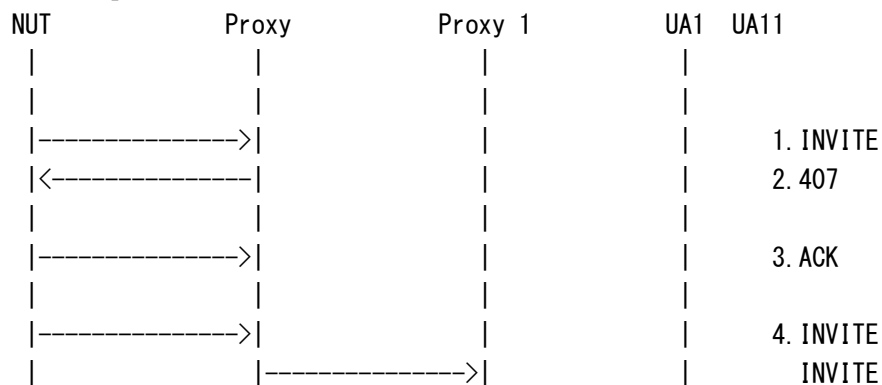
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

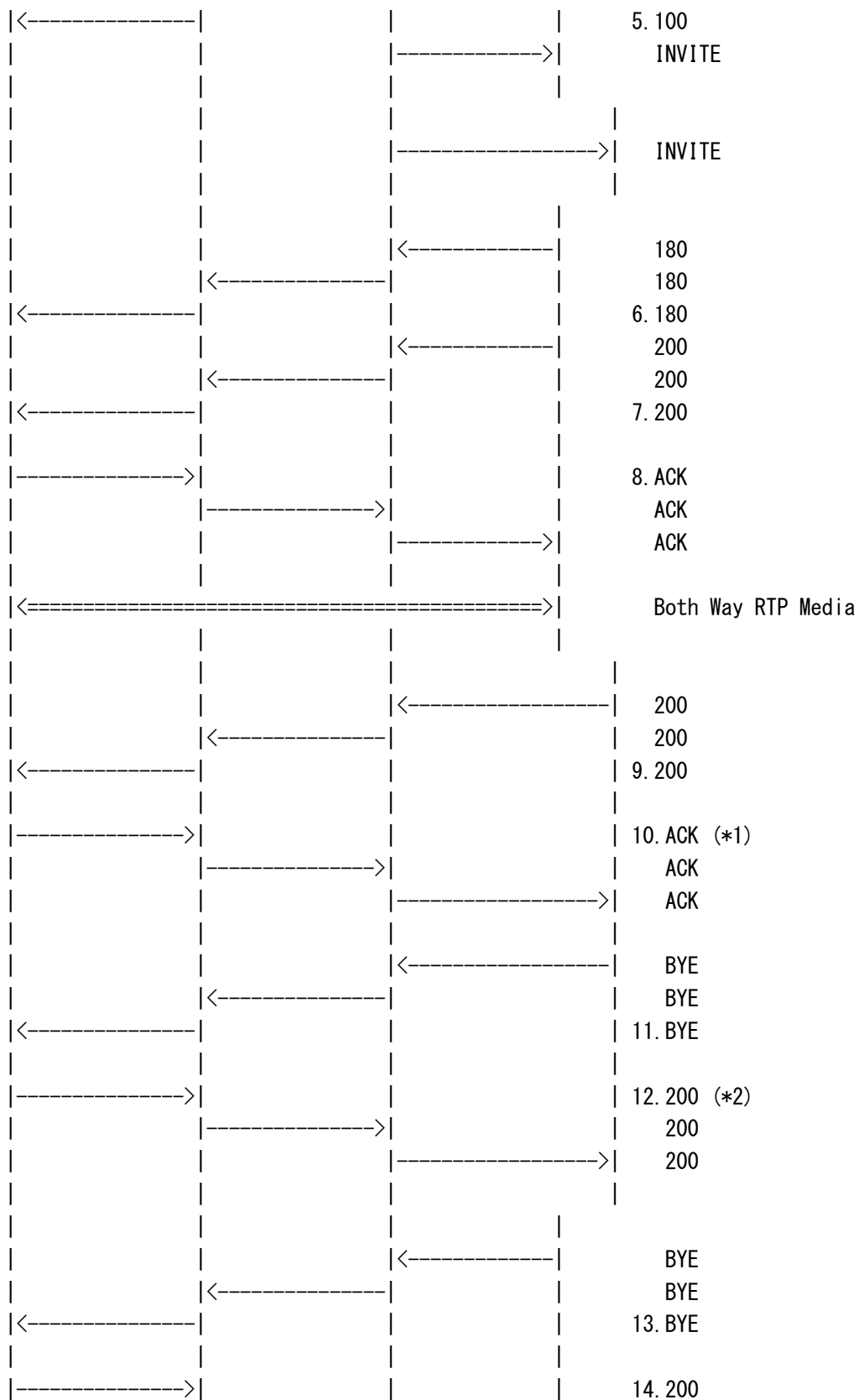
#### [INITIALIZATION]



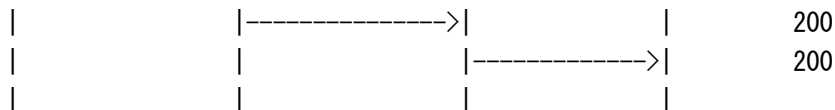
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]









1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
8. Receive ACK.
9. Send 200 OK.
10. Receive ACK. (\*1)
11. Send BYE.
12. Receive 200 OK. (\*2)
13. Send BYE.
14. Receive 200 OK.

**=== Message example ===**

**4. INVITE NUT -> Proxy**

```
INVITE sip:UA1@atlanta.example.com SIP/2.0
(snip)
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 3848276298220188511@under.test.com
CSeq: 2 INVITE
Contact: <sip:NUT@node.under.test.com>
(snip)
```

**6. 180 Ringing Proxy -> NUT**

```
SIP/2.0 180 Ringing
(snip)
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159
Call-ID: 3848276298220188511@under.test.com
Contact: <sip:UA1@client.atlanta.example.com>
CSeq: 2 INVITE
(snip)
```

**7. 200 OK Proxy -> NUT**



SIP/2.0 200 OK  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
(snip)

#### **8. ACK NUT -> Proxy**

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
(snip)

#### **9. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA11@client1.atlanta.example.com>  
(snip)

\* To tag and Contact URI are different from 7.200 OK's.

#### **10. ACK NUT -> Proxy**

ACK sip:UA11@client1.atlanta.example.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
(snip)



#### 11. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

#### 12. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

#### 13. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

#### 14. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

#### [OBSERVABLE RESULTS]

\*1:ACK request from NUT.



As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK  
Request-URI: Must equal the Contact of "9.200" response. [RFC3261-13-18]
- Header fields:  
See generic\_request
- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK
- \* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]
- \* Route  
Must exist. [ORq-2]  
Must contain the Record-Route values of "7.200 OK" in reverse order,  
including all parameters, and the first URI in the route set contains the lr  
parameter. [RFC3261-12-23, 48]
- \* To  
tag-param: Must equal the To tag of "9.200" response. [RFC3261-13-18]
- Bodies:  
See generic\_ACK  
See generic\_2xx-ACK

\*2:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,



- Status-Line:  
See generic\_response
  - \* Status-Code: Must be "200". [RFC3261.22.2.1]
- Header fields:  
See generic\_response
- inside of a dialog
  - \* From  
tag-param: Must equal the From tag of "11.BYE" request.  
[RFC3261-13-18]
  - \* To  
tag-param: Must equal the From tag of "11.BYE" request.  
[RFC3261-13-18]
  - \* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]  
via-received: Must equal the address of UA11. [RFC3261-13-18]

#### [REFERENCE]

[RFC3261-13-16, 17, 18]

##### 13.2.2.4 2xx Responses

Multiple 2xx responses may arrive at the UAC for a single INVITE request due to a forking proxy. Each response is distinguished by the tag parameter in the To header field, and each represents a distinct dialog, with a distinct dialog identifier.

If the dialog identifier in the 2xx response matches the dialog identifier of an existing dialog, the dialog MUST be transitioned to the "confirmed" state, and the route set for the dialog MUST be recomputed based on the 2xx response using the procedures of Section 12.2.1.2. Otherwise, a new dialog in the "confirmed" state MUST be constructed using the procedures of Section 12.1.2.

#### 4.10.5 UA-11-1-5 - Provisional response for the need to ask for an "extension"

**[NAME]**

UA-11-1-5 - Provisional response for the need to ask for an "extension"

**[PURPOSE]**

Verify that a NUT properly processes a provisional response when it needs to ask for an "extension."

**[REQUIREMENT]**

NONE

**[TARGET]**

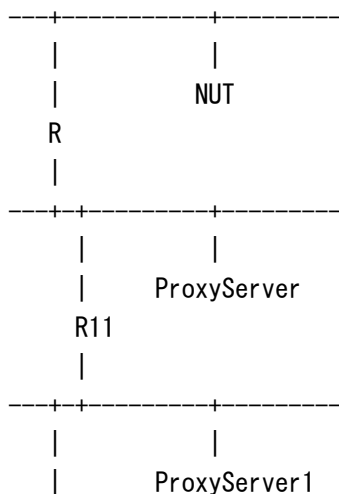
SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

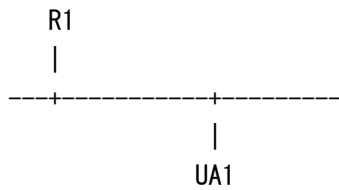
**[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

**[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

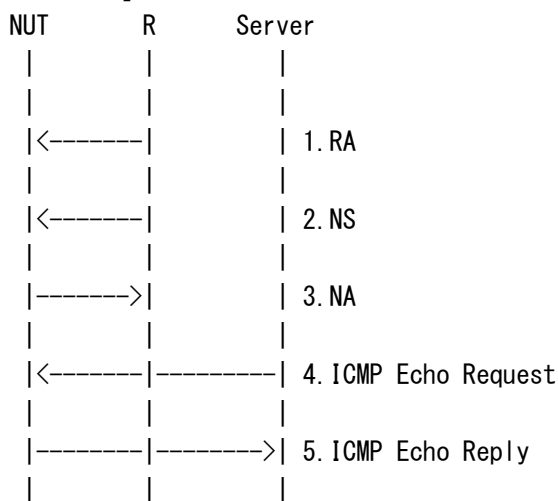
**[TOPOLOGY]**



#### [CONFIGURATION for NUT]

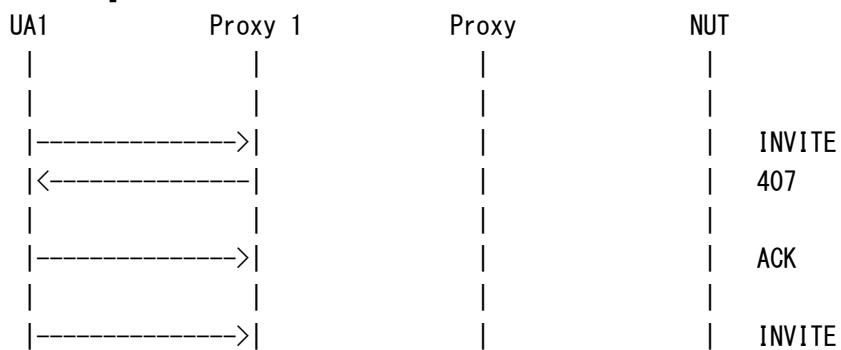
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

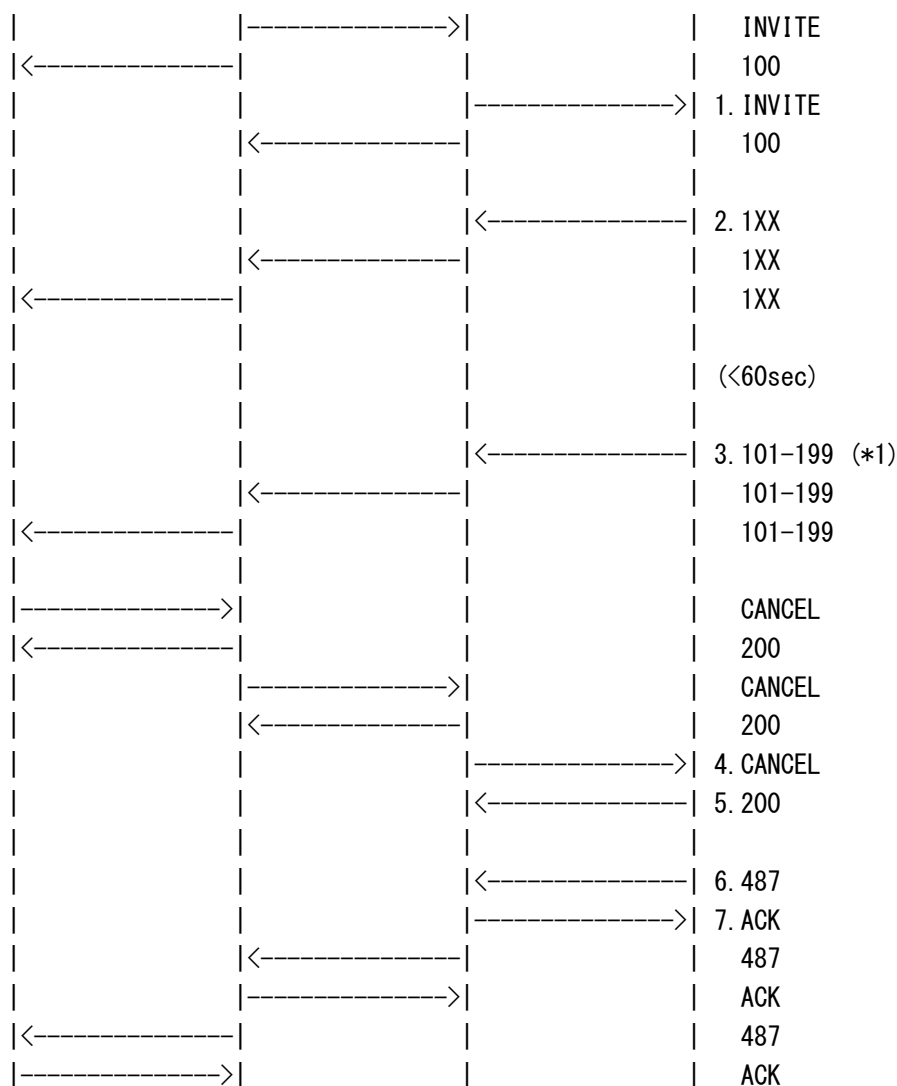
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 101-199. (\*1)
4. Send CANCEL.
5. Receive 200 OK.
6. Receive 487 Request Terminated.
7. Send ACK.

#### [OBSERVABLE RESULTS]

\*1:non-100 provisional response from NUT.

Must send non-100 provisional response within 1 minute. [RFC3261-13-30]

#### [REFERENCE]



[RFC3261-13-30]

#### 13.3.1.1 Progress

If the UAS desires an extended period of time to answer the INVITE, it will need to ask for an "extension" in order to prevent proxies from canceling the transaction. A proxy has the option of canceling a transaction when there is a gap of 3 minutes between responses in a transaction. To prevent cancellation, the UAS **MUST** send a non-100 provisional response at every minute, to handle the possibility of lost provisional responses.

### 4.10.6 UA-11-1-6 - BYE not matching an existing dialog

#### [NAME]

UA-11-1-6 - BYE not matching an existing dialog

#### [PURPOSE]

Verify that a NUT properly processes a BYE request which does not match an existing dialog.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

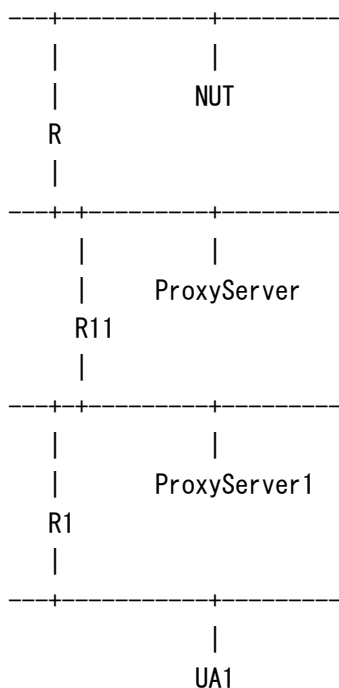
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

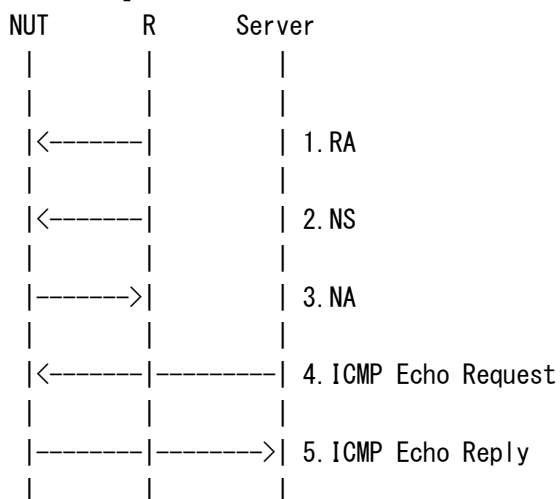
#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

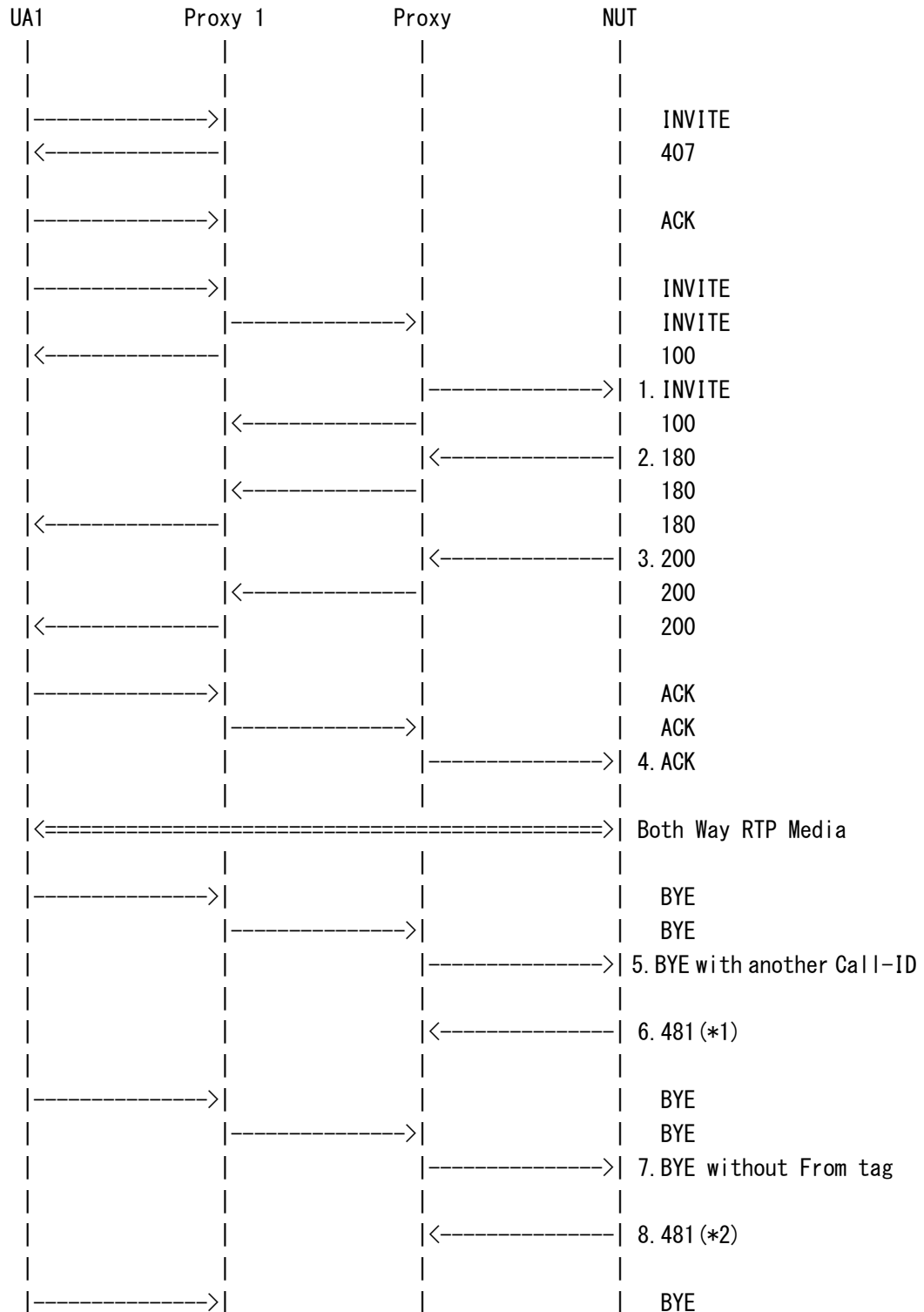
#### [INITIALIZATION]

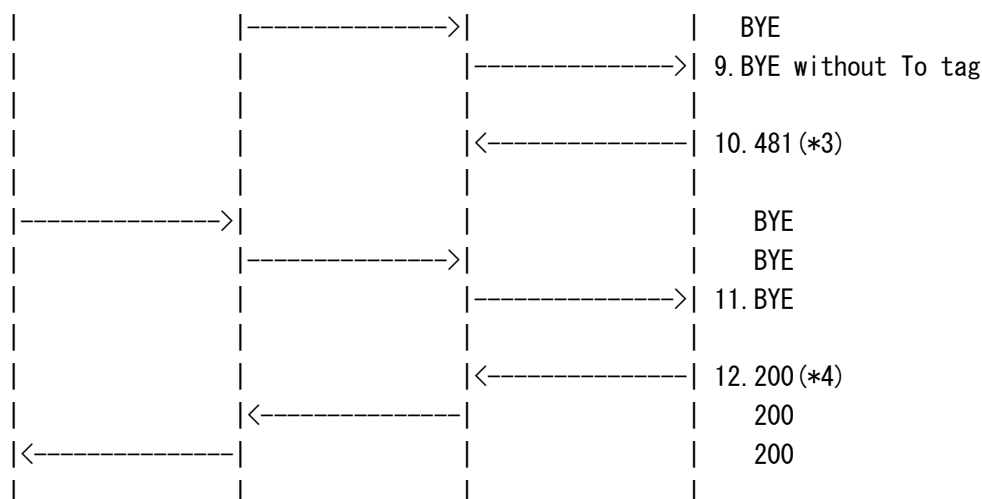


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.

4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send BYE with another Call-ID.
6. Receive 481 Call/Transaction Does Not Exist. (\*1)
7. Send BYE without From tag.
8. Receive 481 Call/Transaction Does Not Exist. (\*2)
9. Send BYE without To tag.
10. Receive 481 Call/Transaction Does Not Exist. (\*3)
11. Send BYE.
12. Receive 200 OK. (\*4)

### === Message example ===

#### 1. INVITE Proxy -> NUT

```
INVITE sip:NUT@node.under.test.com SIP/2.0
(snip)
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
(snip)
```

#### 3. 200 OK NUT -> Proxy

```
SIP/2.0 200 OK
(snip)
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>;tag=314159
```



Call-ID: 3848276298220188511@atlanta.example.com  
(snip)

#### **5.BYE Proxy -> NUT**

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188522@atlanta.example.com  
(snip)

\* Call-ID is different from 1.INVITE's.

#### **7.BYE Proxy -> NUT**

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com  
(snip)

\* No tag in the From header field.

#### **9.BYE Proxy -> NUT**

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 3848276298220188511@atlanta.example.com  
(snip)

\* No tag in the To header field.

#### **11.BYE Proxy -> NUT**

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com



(snip)

## **[OBSERVABLE RESULTS]**

\*1:481 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "481". [RFC3261-15-7]
- Header fields:  
See generic\_response
  - \* Via
    - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
    - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\*2:481 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "481". [RFC3261-15-7]
- Header fields:  
See generic\_response
  - \* Via
    - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
    - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\*3:481 response from NUT.



As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Should be "481". [RFC3261-15-7]
- Header fields:  
See generic\_response
  - \* Via
    - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
    - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\*4:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response
  - \* Status-Code: Must be "200". [RFC3261.22.2.1]
- Header fields:  
See generic\_response
- inside of a dialog
  - \* Via
    - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
    - via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]

## **[REFERENCE]**

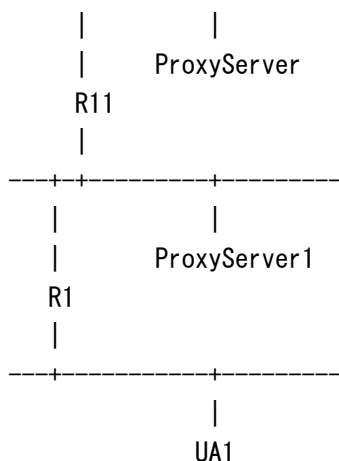
[RFC3261-15-7]

15.1.2 UAS Behavior

A UAS first processes the BYE request according to the general UAS



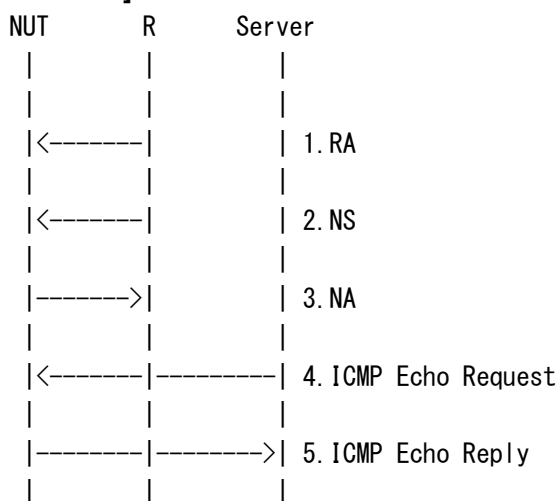




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

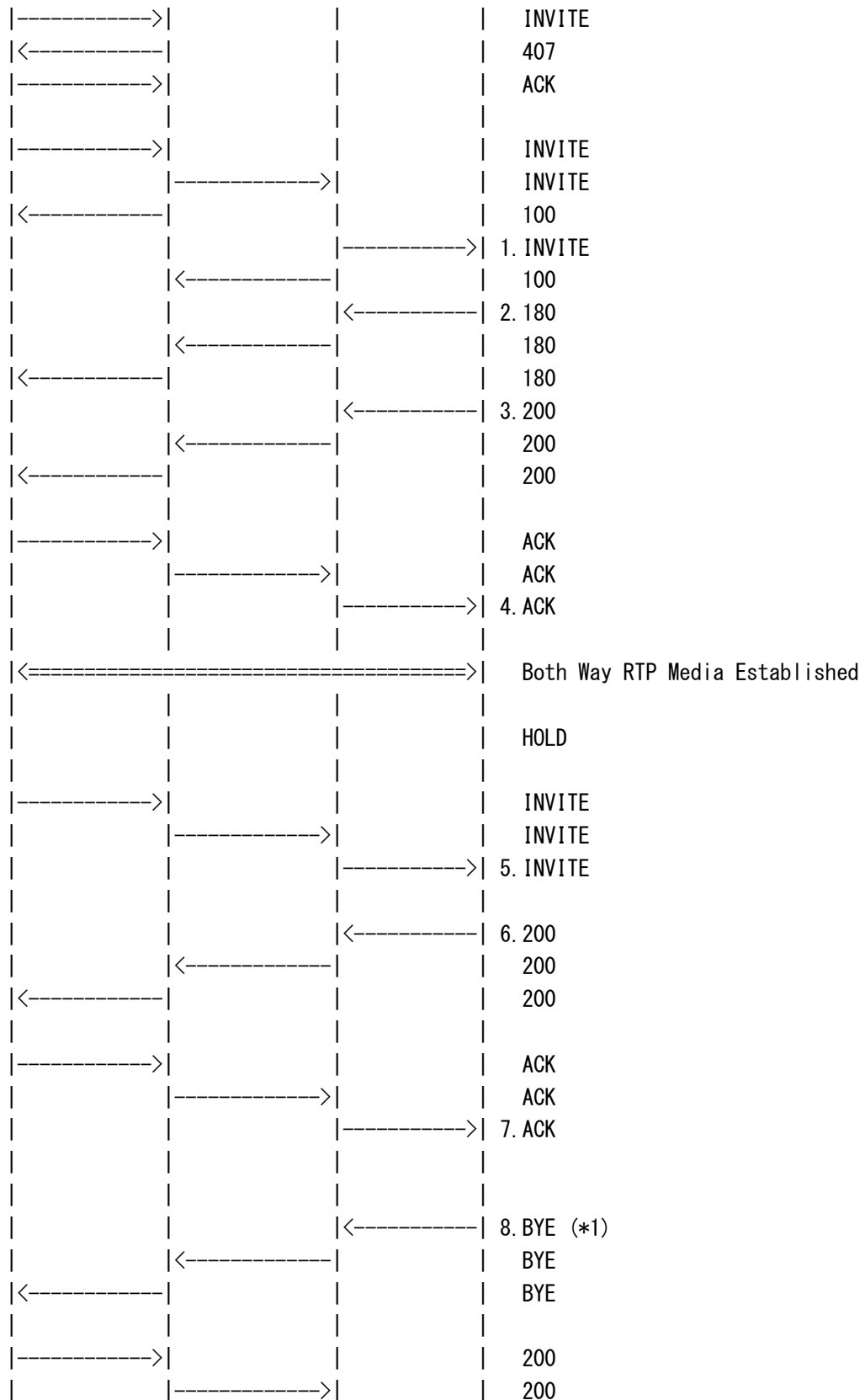
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send INVITE.
6. Receive 200 OK.
7. Send ACK.
8. Receive BYE. (\*1)
9. Send 200 OK.

**=== Message example ===**

**1. INVITE Proxy -> NUT**

(snip)

Record-Route: <sip:ss.under.test.com;lr>,

<sip:ss1.atlanta.example.com;lr>

Contact: <sip:UA1@client.atlanta.example.com>

(snip)

**5. INVITE Proxy -> NUT**

(snip)

Record-Route: <sip:ss1.atlanta.example.com;lr>,

<sip:ss.under.test.com;lr>

Contact: <sip:UA1@client.atlanta.example.com>

(snip)

\* Tester sends re-INVITE, but Record-Route is different from Initial-Invite.

**6. 200 OK NUT -> Proxy**

SIP/2.0 200 OK

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1

;received=3ffe:501:ffff:50::50

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9

;received=3ffe:501:ffff:1::1



From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=NUT 2890844527 2890844528 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=recvonly

\* It is undefined whether this response includes Record-Route header field or not.

## 8. BYE NUT -> Proxy

BYE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bo4  
Route: <sip:ss.under.test.com;lr>,  
      <sip:ss1.atlanta.example.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 3 BYE  
Content-Length: 0

\* Route header field is generated from the route set that was determined when the dialog was created.

## [OBSERVABLE RESULTS]

\*1:BYE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,



- Request-Line:
  - See generic\_request
  - See generic\_BYE
  - Request-URI: Must be the URI of Contact in "7.200" response. [RFC3261-12-47]
- Header fields:
  - See generic\_request
- inside of a dialog
  - See generic\_BYE
- \* To
  - tag-param: Must equal that contained in the To header field of "7.200" response. [RFC3261-12-35]
- \* From
  - tag-param: Must equal that contained in the From header field of "1.INVITE". [RFC3261-12-37]
- \* Route
  - Must include a Route header field. [RFC3261-12-48]
  - route-param: Must contain the route set values in order, including all parameters.[RFC3261-12-48]
- Bodies:
  - See generic\_BYE
- (Special mention matter)
  - automated generation of re-INVITE or BYE is not recommended. [RFC3261-14-1]
  - Must not initiate a new INVITE transaction within a dialog while another INVITE transaction is in progress in either direction. [RFC3261-14-4]

## [REFERENCE]

[rfc3261]

### 12.2 Requests within a Dialog

Once a dialog has been established between two UAs, either of them MAY initiate new transactions as needed within the dialog. The UA sending the request will take the UAC role for the transaction. The UA receiving the request will take the UAS role. Note that these may be different roles than the UAs held during the transaction that



established the dialog.

Requests within a dialog MAY contain Record-Route and Contact header fields. However, these requests do not cause the dialog's route set to be modified, although they may modify the remote target URI. Specifically, requests that are not target refresh requests do not modify the dialog's remote target URI, and requests that are target refresh requests do. For dialogs that have been established with an

INVITE, the only target refresh request defined is re-INVITE (see Section 14). Other extensions may define different target refresh requests for dialogs established in other ways.

Note that an ACK is NOT a target refresh request.

Target refresh requests only update the dialog's remote target URI, and not the route set formed from the Record-Route. Updating the latter would introduce severe backwards compatibility problems with RFC 2543-compliant systems.

#### 4.10.8 UA-11-1-8 - Returning of correct Record-Route parameter

##### [NAME]

UA-11-1-8 - Returning of correct Record-Route parameter

##### [PURPOSE]

Verify that a NUT properly returns a correct Record-Route parameter .

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

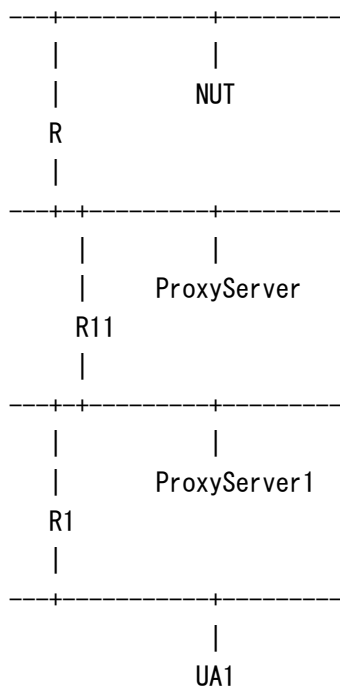
##### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

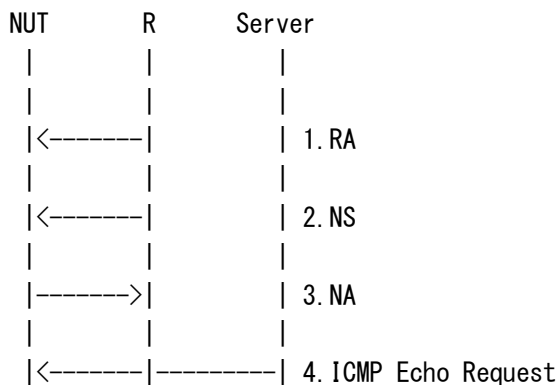
#### [TOPOLOGY]

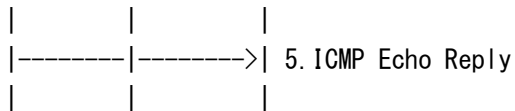


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

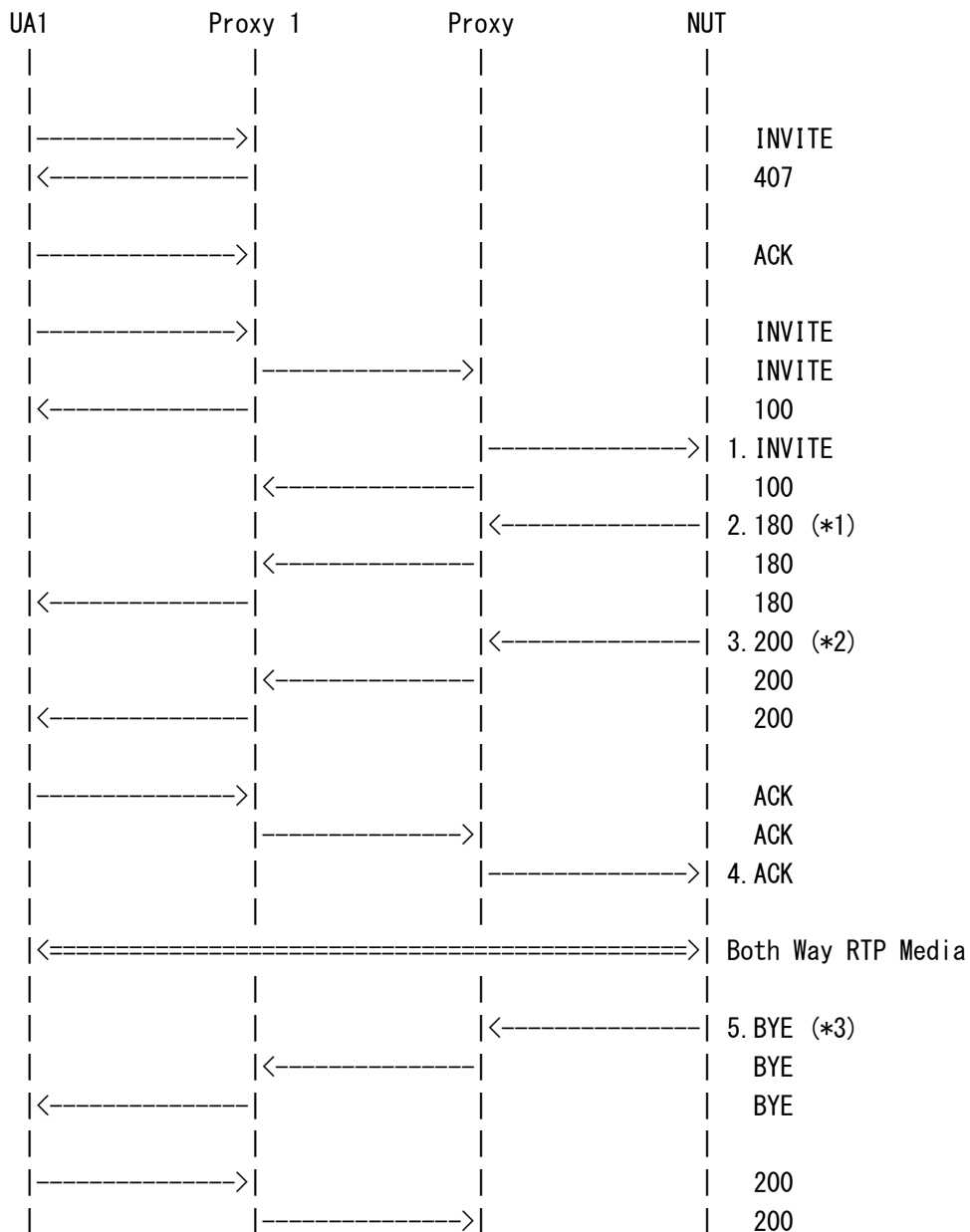
#### [INITIALIZATION]





1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







		----->	6. 200

1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
5. Receive BYE. (\*3)
6. Send 200 OK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

(snip)

Record-Route: <sip:ss.under.test.com;lr;unknown>;unknown2,  
 <sip:ss1.atlanta.example.com;lr;unknown3>;unknown4  
 Contact: <sip:UA1@client.atlanta.example.com>

(snip)

\* Record-Route header field has unknown parameters.

#### 5. BYE NUT -> Proxy

BYE sip:UA1@client.atlanta.example.com SIP/2.0  
 Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKnashds7  
 Max-Forwards: 70  
 Route: <sip:ss.under.test.com;lr;unknown>;unknown2,  
 <sip:ss1.atlanta.example.com;lr;unknown3>;unknown4  
 From: NUT <sip:NUT@under.test.com>;tag=314159  
 To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
 Call-ID: 3848276298220188511@atlanta.example.com  
 CSeq: 1 BYE  
 Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:180 response from NUT. [Optional]

As a SIP Message,  
 See generic\_message

As a SIP response,

- Status-Line:



See generic\_response

Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:

See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\*2:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

See generic\_200-for-INVITE

Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:

See generic\_response

- outside of a dialog

See generic\_200-for-INVITE

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was



received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values.  
[RFC3261-12-3]

- Bodies:

See generic\_200-for-INVITE

See generic\_SDP

\*3:BYE request from NUT.

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request.  
[RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "3.200" response.  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all



parameters. [RFC3261-12-48]

- Bodies:  
See generic\_BYE

#### [REFERENCE]

[RFC3261-12-2]

##### 12.1.1 UAS behavior

When a UAS responds to a request with a response that establishes a dialog (such as a 2xx to INVITE), the UAS MUST copy all Record-Route header field values from the request into the response (including the URIs, URI parameters, and any Record-Route header field parameters, whether they are known or unknown to the UAS) and MUST maintain the order of those values.

#### 4.10.9 UA-11-1-9 - Proper processing upon receipt of multiple 18x from multiple downstream branches

#### [NAME]

UA-11-1-9 - Proper processing upon receipt of multiple 18x from multiple downstream branches

#### [PURPOSE]

Verify that a NUT properly processes when receiving multiple 18x response from multiple downstream branches.

#### [REQUIREMENT]

Only when a NUT supports multiple response.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

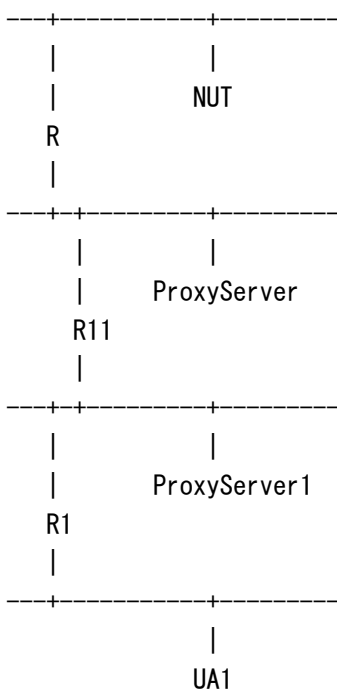
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA11(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
UA11(Contact)	sip:UA11@client1.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
UA11(IPv6)	3ffe:501:ffff:1::11/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

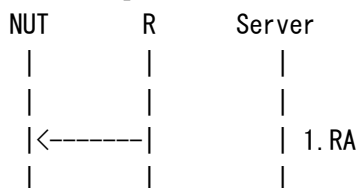
# [TOPOLOGY]

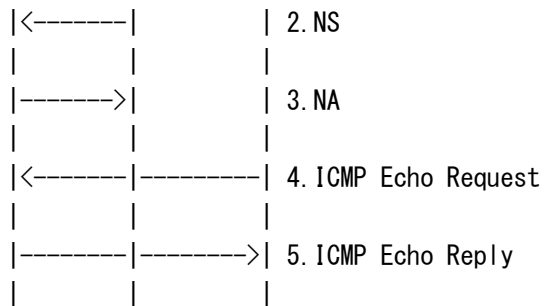


# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

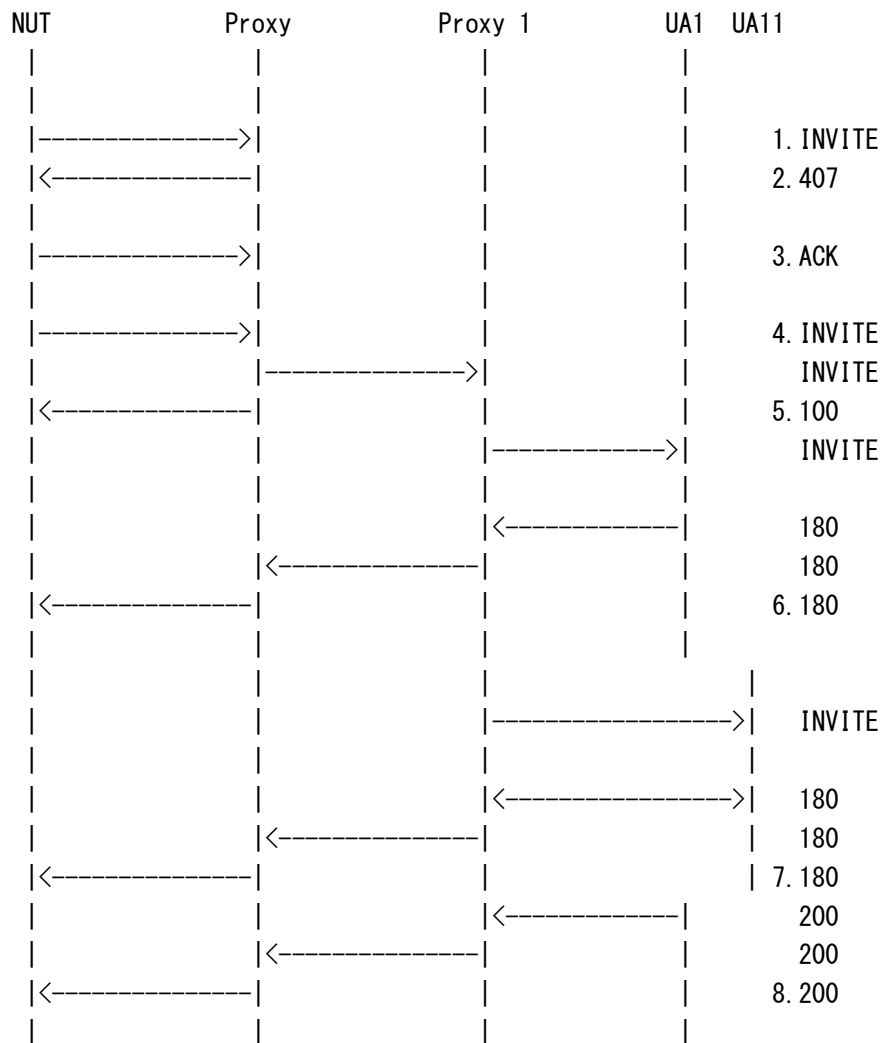
# [INITIALIZATION]

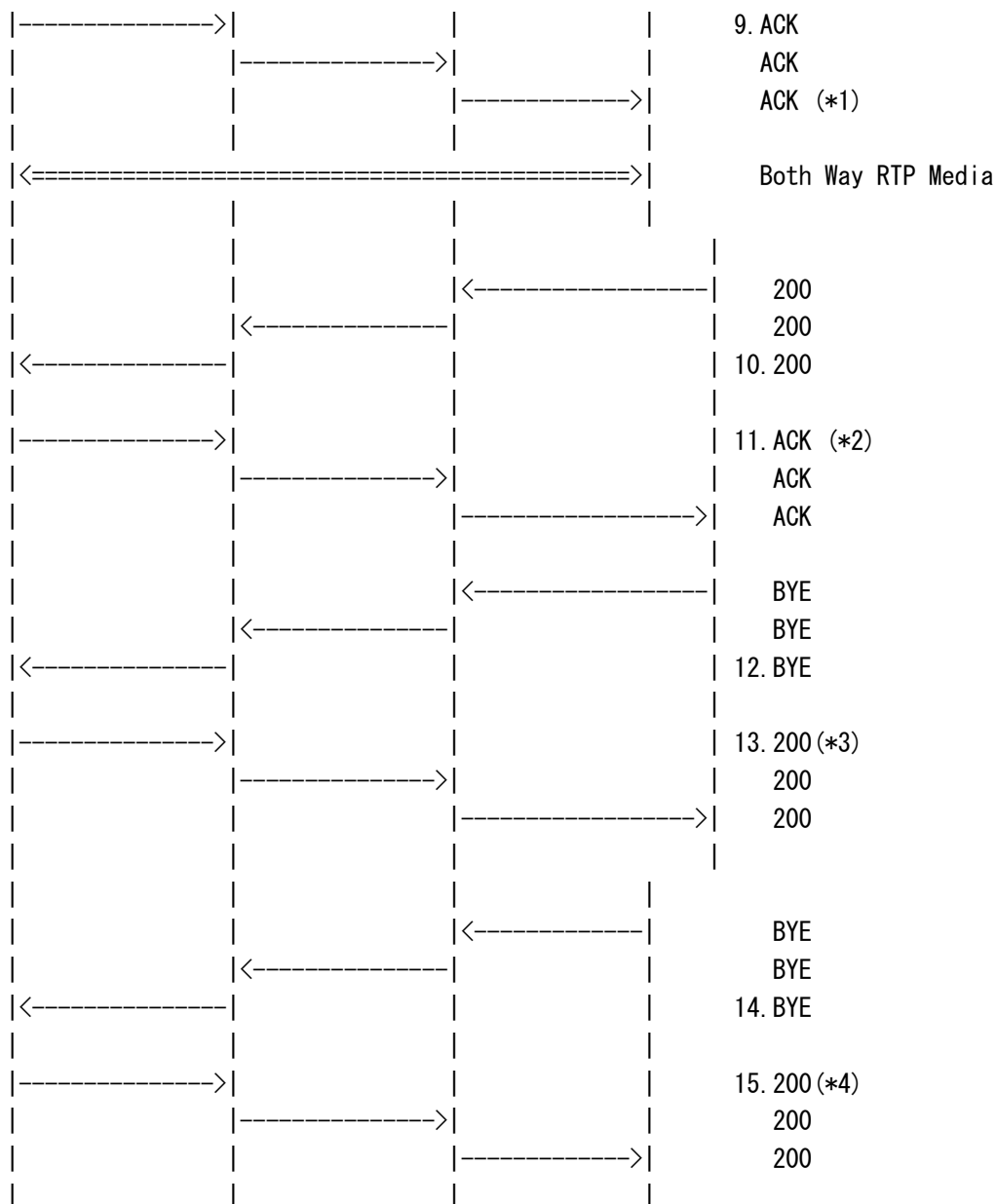




1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 180 Ringing.
8. Send 200 OK.
9. Receive ACK. (\*1)
10. Send 200 OK.



11. Receive ACK. (\*2)
12. Send BYE.
13. Receive 200. (\*3)
14. Send BYE.
15. Receive 200. (\*4)

=== Message example ===

#### 4. INVITE NUT -> Proxy

```
INVITE sip:UA1@atlanta.example.com SIP/2.0
(snip)
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 3848276298220188511@under.test.com
CSeq: 2 INVITE
Contact: <sip:NUT@node.under.test.com>
(snip)
```

#### 6. 180 Ringing Proxy -> NUT

```
SIP/2.0 180 Ringing
(snip)
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159
Call-ID: 3848276298220188511@under.test.com
Contact: <sip:UA1@client.atlanta.example.com>
CSeq: 2 INVITE
(snip)
```

#### 7. 180 Ringing Proxy -> NUT

```
SIP/2.0 180 Ringing
(snip)
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>;tag=98765
Call-ID: 3848276298220188511@under.test.com
Contact: <sip:UA1@client.atlanta.example.com>
CSeq: 2 INVITE
(snip)
```

#### 8. 200 OK Proxy -> NUT

```
SIP/2.0 200 OK
```





(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
(snip)

#### **9. ACK NUT -> Proxy**

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
(snip)

\* To tag equals that of 8.200 (not 7.180)

#### **10. 200 OK Proxy -> NUT**

SIP/2.0 200 OK  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA11@client1.atlanta.example.com>  
(snip)

\* To tag and Contact URI are different from 8.200 OK's.

#### **11. ACK NUT -> Proxy**

ACK sip:UA11@client1.atlanta.example.com SIP/2.0  
(snip)  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
(snip)



\* To tag equals that of 10.200 (not 8.200)

## 12. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

## 13. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=98765  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

## 14. BYE Proxy -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)

## 15. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
(snip)  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 1 BYE  
(snip)



## [OBSERVABLE RESULTS]

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK  
Request-URI: Must equal the Contact of "8.200" response. [RFC3261-13-18]
- Header fields:  
See generic\_request
- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK
- \* Proxy-Authorization  
Must exist. [RFC3261.22.3],[RFC3261-22-22]  
Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]
- \* Route  
Must exist. [ORq-2]  
Must contain the Record-Route values of "8.200 OK" in reverse order,  
including all parameters, and the first URI in the route set contains the lr  
parameter. [RFC3261-12-23, 48]
- \* To  
tag-param: Must equal the To tag of "8.200" response. [RFC3261-13-18]
- Bodies:  
See generic\_ACK  
See generic\_2xx-ACK

\*2:ACK request from NUT.

As a SIP Message,



See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

Request-URI: Must equal the Contact of "10.200" response. [RFC3261-13-18]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "8.200 OK" in reverse order,  
including all parameters, and the first URI in the route set contains the lr  
parameter. [RFC3261-12-23, 48]

\* To

tag-param: Must equal the To tag of "10.200" response. [RFC3261-13-18]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*3:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,



- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261.22.2.1]

- Header fields:

See generic\_response

- inside of a dialog

\* From

tag-param: Must equal the From tag of "12.BYE" request.  
[RFC3261-13-18]

\* To

tag-param: Must equal the From tag of "12.BYE" request.  
[RFC3261-13-18]

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter  
contains a domain name. [RFC3261-18-27]

via-received: MUST contain the source address from which the packet was  
received. [RFC3261-18-28]

via-received: Must equal the address of UA11. [RFC3261-13-18]

\*4:200 response from NUT.

As a SIP Message,

See generic\_message

As a SIP response,

- Status-Line:

See generic\_response

\* Status-Code: Must be "200". [RFC3261.22.2.1]

- Header fields:

See generic\_response

- inside of a dialog

\* From

tag-param: Must equal the From tag of "14.BYE" request.  
[RFC3261-13-18]



- \* To  
tag-param: Must equal the From tag of "14.BYE" request.  
[RFC3261-13-18]
- \* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: MUST contain the source address from which the packet was received. [RFC3261-18-28]  
via-received: Must equal the address of UA1. [RFC3261-13-18]

#### [REFERENCE]

[RFC3261-13-16, 17, 18]  
13.2.2.4 2xx Responses

Multiple 2xx responses may arrive at the UAC for a single INVITE request due to a forking proxy. Each response is distinguished by the tag parameter in the To header field, and each represents a distinct dialog, with a distinct dialog identifier.

If the dialog identifier in the 2xx response matches the dialog identifier of an existing dialog, the dialog MUST be transitioned to the "confirmed" state, and the route set for the dialog MUST be recomputed based on the 2xx response using the procedures of Section 12.2.1.2. Otherwise, a new dialog in the "confirmed" state MUST be constructed using the procedures of Section 12.1.2.

#### 4.10.10 UA-11-1-10 - Failed re-INVITE not changing the dialog

##### [NAME]

UA-11-1-10 - Failed re-INVITE not changing the dialog

##### [PURPOSE]

Verify that a NUT doesn't change the dialog when failure of re-INVITE.

##### [REQUIREMENT]

Only when a NUT supports hold using re-INVITE.

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

##### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
----------	------------------------

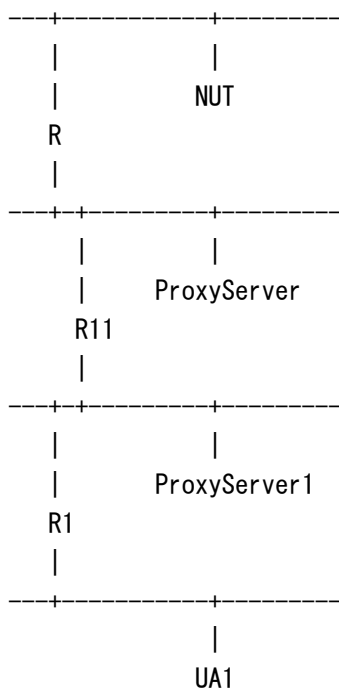


NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

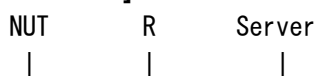
#### [TOPOLOGY]

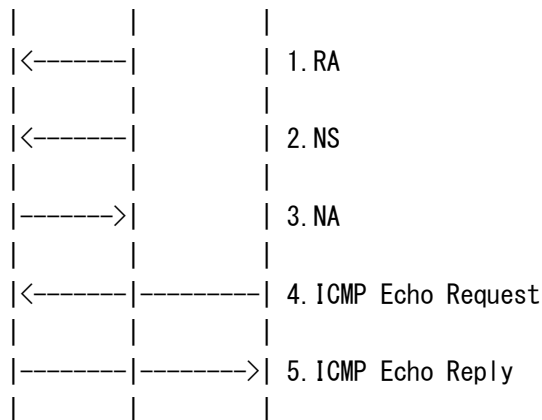


#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

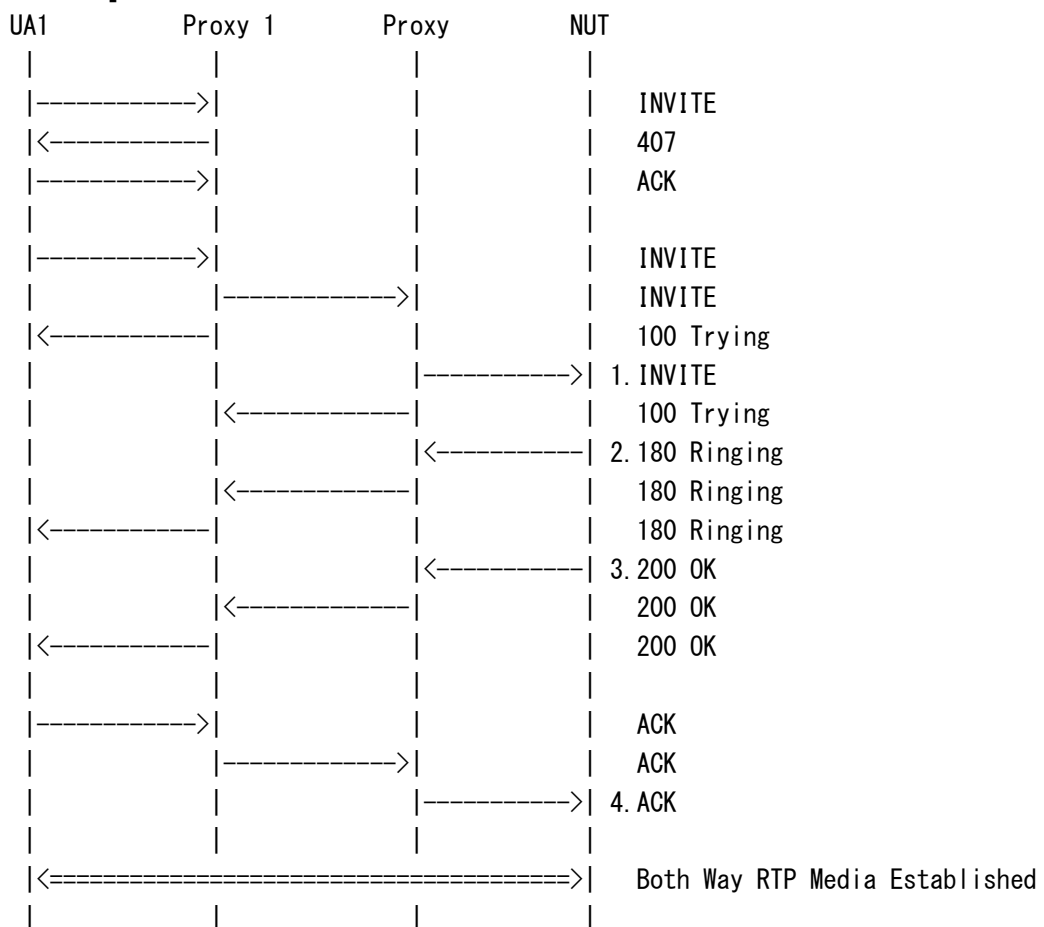
#### [INITIALIZATION]



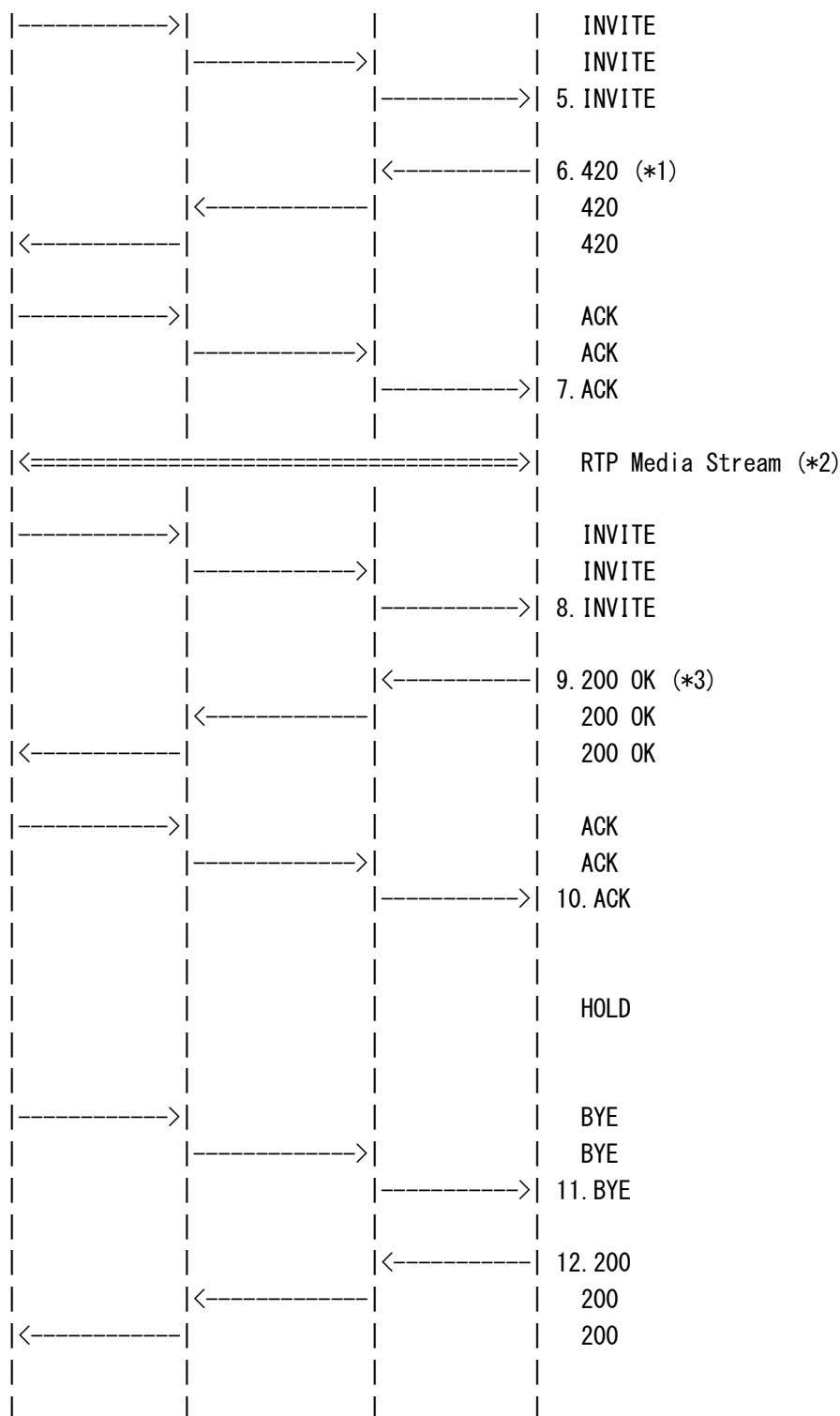


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]







1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.



4. Send ACK.
5. Send INVITE.
6. Receive 420 Bad Extension. (\*1)
7. Send ACK.  
(\*2)
8. Send INVITE.
9. Receive 200 OK. (\*3)
10. Send ACK.
11. Send BYE.
12. Receive 200 OK.

**=== Message example ===**

**5. INVITE Proxy -> NUT**

```
INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>;tag=314159
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com
CSeq: 2 INVITE
Require: foo
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 149

v=0
o=UA1 2890844527 2890844528 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly
```

\* Tester sends re-INVITE, including invalid Require header field.

**6. 420 NUT -> Proxy**



SIP/2.0 420 Bad Extension

Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 2 INVITE  
Unsupported: foo  
Content-Length: 0

## 7. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 2 ACK  
Content-Length: 0

## 8. INVITE Proxy -> NUT

INVITE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 3 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>



Content-Type: application/sdp  
Content-Length: 149

v=0  
o=UA1 2890844527 2890844528 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=sendonly

#### 9. 200 OK NUT -> Proxy

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:50::50  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 3 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 150

v=0  
o=NUT 2890844527 2890844528 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=recvonly

#### 10. ACK Proxy -> NUT

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1



;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 3 ACK  
Content-Length: 0

### [OBSERVABLE RESULTS]

\*1:420 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "420". [RFC3261-21-17]

- Header fields:  
See generic\_response

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Unsupported:  
Must exist. [RFC3261-21-12]  
option-tag: Must include "foo". [RFC3261-21-17]

\*2:RTP must exist. (Must not succeed in HOLD) [RFC3261-8-64]

\*3:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,



- Status-Line:
  - See generic\_response
  - See generic\_200-for-INVITE
  - Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]
- Header fields:
  - See generic\_response
- inside of a dialog
  - See generic\_200-for-INVITE
- \* Via
  - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
  - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]
- \* Record-Route
  - Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]
  - rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]
- Bodies:
  - See generic\_200-for-INVITE
  - See generic\_SDP

## [REFERENCE]

[RFC3261-8-63, 64]

### 8.2 UAS Behavior

When a request outside of a dialog is processed by a UAS, there is a set of processing rules that are followed, independent of the method. Section 12 gives guidance on how a UAS can tell whether a request is inside or outside of a dialog.

Note that request processing is atomic. If a request is accepted, all state changes associated with it **MUST** be performed. If it is rejected, all state changes **MUST NOT** be performed.

## 12.2.2 UAS Behavior

Requests sent within a dialog, as any other requests, are atomic. If a particular request is accepted by the UAS, all the state changes associated with it are performed. If the request is rejected, none of the state changes are performed.

[RFC3261-21-17]

### 21.4.15 420 Bad Extension

The server did not understand the protocol extension specified in a Proxy-Require (Section 20.29) or Require (Section 20.32) header field. The server **MUST** include a list of the unsupported extensions in an Unsupported header field in the response. UAC processing of this response is described in Section 8.1.3.5.

## 4.10.11 UA-11-1-11 - Construction of a correct Route set

### [NAME]

UA-11-1-11 - Construction of a correct Route set

### [PURPOSE]

Verify that a NUT properly copies Route set including parameters.

### [REQUIREMENT]

NONE

### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

### [PARAMETER]

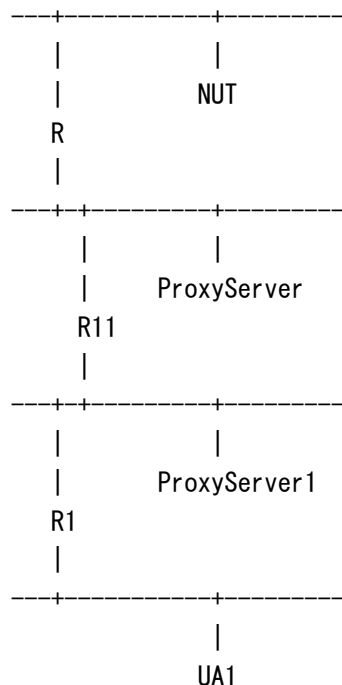
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr;unknown
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr;unknown2

### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64

UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

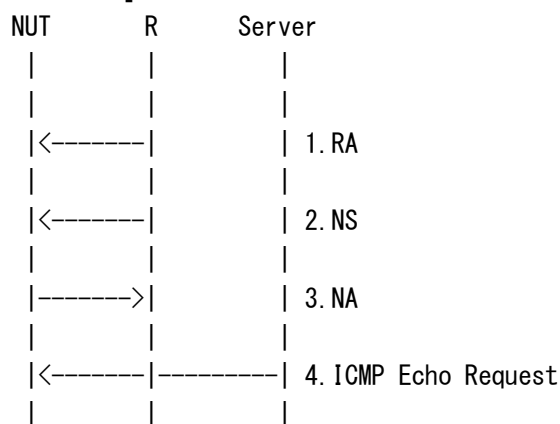
# [TOPOLOGY]



# [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr;unknown
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

# [INITIALIZATION]

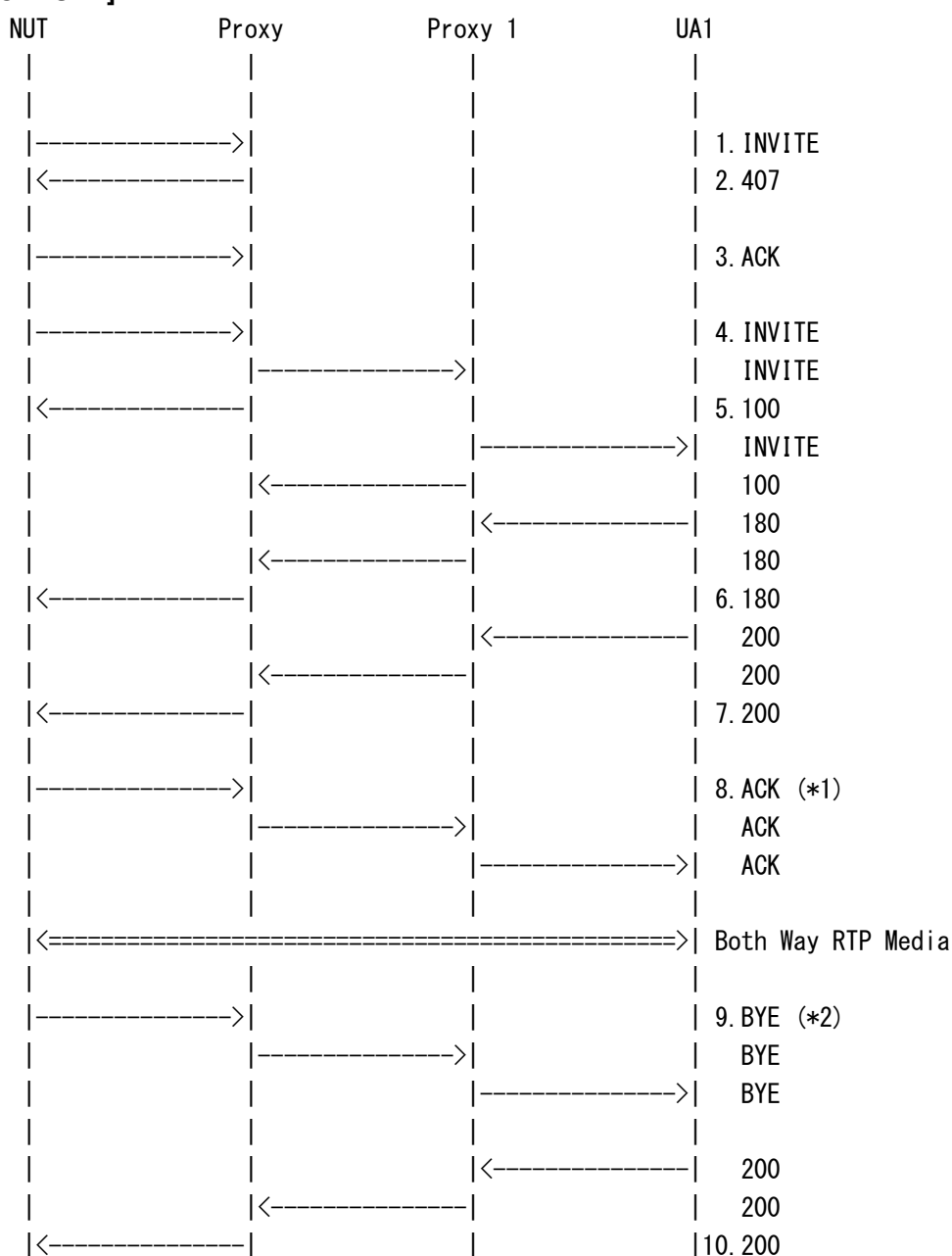




|-----|----->| 5. ICMP Echo Reply  
| | |

1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]



1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying
6. Send 180 Ringing
7. Send 200 OK
8. Receive ACK (\*1)
9. Receive BYE (\*2)
10. Send 200

**=== Message example ===**

**4. INVITE NUT -> Proxy**

```
INVITE sip:UA1@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
Proxy-Authorization: Digest username="NUT",
  realm="under.test.com",
  nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",
  qop=auth, nc=00000004, cnonce="6f54a149",
  uri="sip:UA1@atlanta.example.com",
  response="b51e504e73af54829e4f2bd7f8dc4654"
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>
Call-ID: 3848276298220188511@under.test.com
CSeq: 2 INVITE
Contact: <sip:NUT@node.under.test.com>
Content-Type: application/sdp
Content-Length: 151
```

```
v=0
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X
s=-
c=IN IP6 3ffe:501:ffff:5::X
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

- /\* Proxy accepts the credentials and forwards the INVITE to Proxy
2. Client for NUT prepares to receive data on port 49172 from the



network. \*/

## 5. 100 Trying Proxy -> NUT

SIP/2.0 100 Trying

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

## 6. 180 Ringing Proxy -> NUT

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr;unknown2>,  
<sip:ss.under.test.com;lr;unknown>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
Contact: <sip:UA1@client.atlanta.example.com>  
CSeq: 2 INVITE  
Content-Length: 0

\* Record-Route header field has "unknown2" and "unknown" parameter.

## 7. 200 OK Proxy -> NUT

SIP/2.0 200 OK

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr;unknown2>,  
<sip:ss.under.test.com;lr;unknown>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 147



v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 8. ACK NUT -> Proxy

ACK sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b76  
Max-Forwards: 70  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="1cec4341ae6cbe5a359ea9c8e88df84f", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
Route: <sip:ss.under.test.com;lr;unknown>,  
<sip:ss1.atlanta.example.com;lr;unknown2>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 ACK  
Content-Length: 0

## 9. BYE NUT -> Proxy

BYE sip:UA1@client.atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b77  
Max-Forwards: 70  
Route: <sip:ss.under.test.com;lr;unknown>,  
<sip:ss1.atlanta.example.com;lr;unknown2>  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 1 BYE  
Content-Length: 0

## 10.200 OK Proxy -> NUT



SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b77  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=314159  
To: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 1 BYE  
Content-Length: 0

#### **[OBSERVABLE RESULTS]**

\*1:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_2xx-ACK

- Header fields:  
See generic\_request

- inside of a dialog  
See generic\_ACK  
See generic\_2xx-ACK

\* Proxy-Authorization  
Must exist. [RFC3261-22.3],[RFC3261-22-22]  
Must be the same as that of \*3 Proxy-Authorization. [RFC3261-13-22]

\* Route  
Must exist. [ORq-2]  
Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:  
See generic\_ACK  
See generic\_2xx-ACK



\*3:BYE request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_BYE

Request-URI: Must be the URI of Contact in "1.INVITE" request.  
[RFC3261-12-47]

- Header fields:

See generic\_request

- inside of a dialog

See generic\_BYE

\* To

tag-param: Must equal that contained in the From header field of "1.INVITE".  
[RFC3261-12-35]

\* From

tag-param: Must equal that contained in the To header field of "3.200" response.  
[RFC3261-12-37]

\* Route

Must include a Route header field. [RFC3261-12-48]

route-param: Must contain the route set values in order, including all  
parameters. [RFC3261-12-48]

- Bodies:

See generic\_BYE

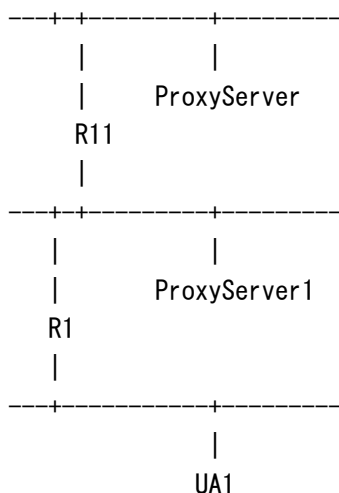
## [REFERENCE]

[RFC3261-12-23]

### 12.1.2 UAC Behavior

The route set MUST be set to the list of URIs in the Record-Route header field from the response, taken in reverse order and preserving all URI parameters. If no Record-Route header field is present in the response, the route set MUST be set to the empty set. This route set, even if empty, overrides any pre-existing route set for future

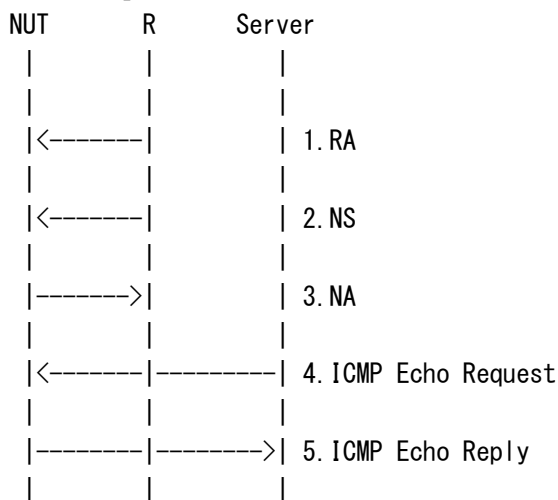




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

#### [INITIALIZATION]

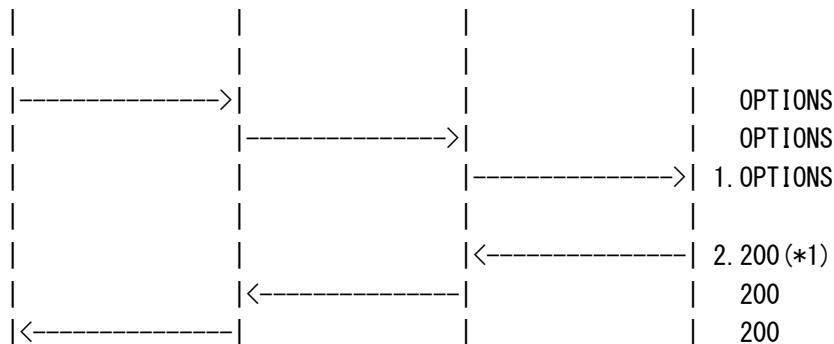


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

UA1                      Proxy 1                      Proxy                      NUT





1. Send OPTIONS.
2. Receive 200 OK. (\*1)

### === Message example ===

#### 1.OPTIONS Proxy -> NUT

```

OPTIONS sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 OPTIONS
Contact: <sip:UA1@client.atlanta.example.com>
Accept: application/sdp
Content-Length: 0
  
```

### [OBSERVABLE RESULTS]

\*1:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response



Status-Code: Must be "200". [RFC3261-11-1,3]

- Header fields:

See generic\_response

- inside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq-2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\* Allow

Should exist. [RFC3261-11-4]

\* Accept

Should exist. [RFC3261-11-4]

\* Accept-Encoding

Should exist. [RFC3261-11-4]

\* Accept-Language

Should exist. [RFC3261-11-4]

\* Supported

Should exist. [RFC3261-11-4]

- Bodies:

See generic\_SDP

Should exist. [RFC3261-11-6]

**[REFERENCE]**

[RFC3261-11-3, 4, 5]

11 Querying for Capabilities



The SIP method OPTIONS allows a UA to query another UA or a proxy server as to its capabilities. This allows a client to discover information about the supported methods, content types, extensions, codecs, etc. without "ringing" the other party. For example, before a client inserts a Require header field into an INVITE listing an option that it is not certain the destination UAS supports, the client can query the destination UAS with an OPTIONS to see if this option is returned in a Supported header field. All UAs MUST support the OPTIONS method.

## 11.2 Processing of OPTIONS Request

The response to an OPTIONS is constructed using the standard rules for a SIP response as discussed in Section 8.2.6. The response code chosen MUST be the same that would have been chosen had the request been an INVITE. That is, a 200 (OK) would be returned if the UAS is ready to accept a call, a 486 (Busy Here) would be returned if the UAS is busy, etc. This allows an OPTIONS request to be used to determine the basic state of a UAS, which can be an indication of whether the UAS will accept an INVITE request.

(snip)

Allow, Accept, Accept-Encoding, Accept-Language, and Supported header fields SHOULD be present in a 200 (OK) response to an OPTIONS request. If the response is generated by a proxy, the Allow header field SHOULD be omitted as it is ambiguous since a proxy is method agnostic. Contact header fields MAY be present in a 200 (OK) response and have the same semantics as in a 3xx response. That is, they may list a set of alternative names and methods of reaching the user. A Warning header field MAY be present.

### 4.11.2 UA-12-1-2 - Receipt of OPTIONS within a dialog

#### [NAME]

UA-12-1-2 – Receipt of OPTIONS within a dialog

#### [PURPOSE]

Verify that a NUT properly receives an OPTIONS request within a dialog.

#### [REQUIREMENT]

Only when a NUT supports the function of OPTIONS request.

# **[TARGET]**

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

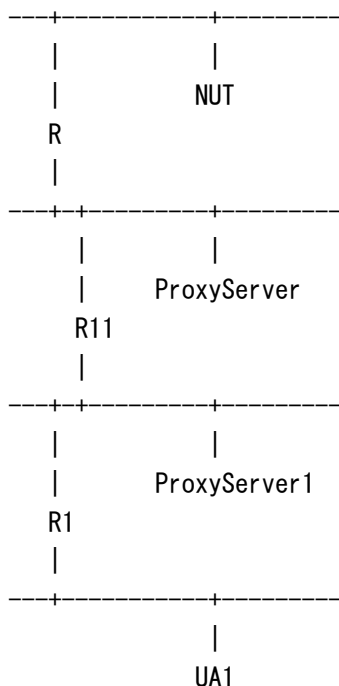
# **[PARAMETER]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

# **[ADDRESS]**

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

# **[TOPOLOGY]**

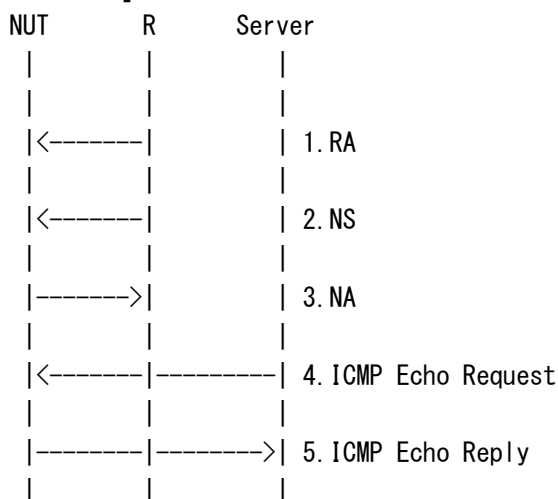


# **[CONFIGURATION for NUT]**

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com

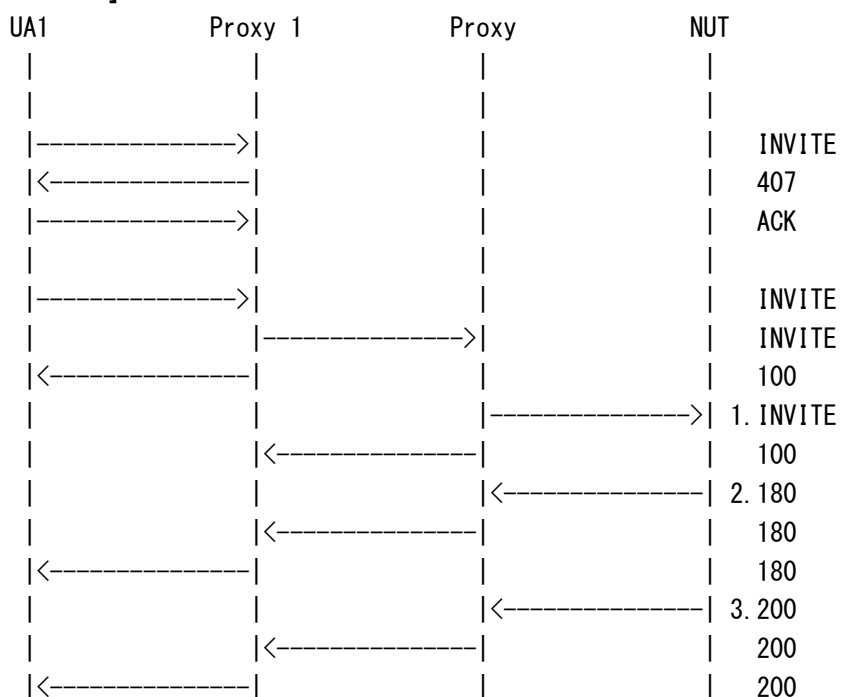
ProxyServer	sip:ss.under.test.com:lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

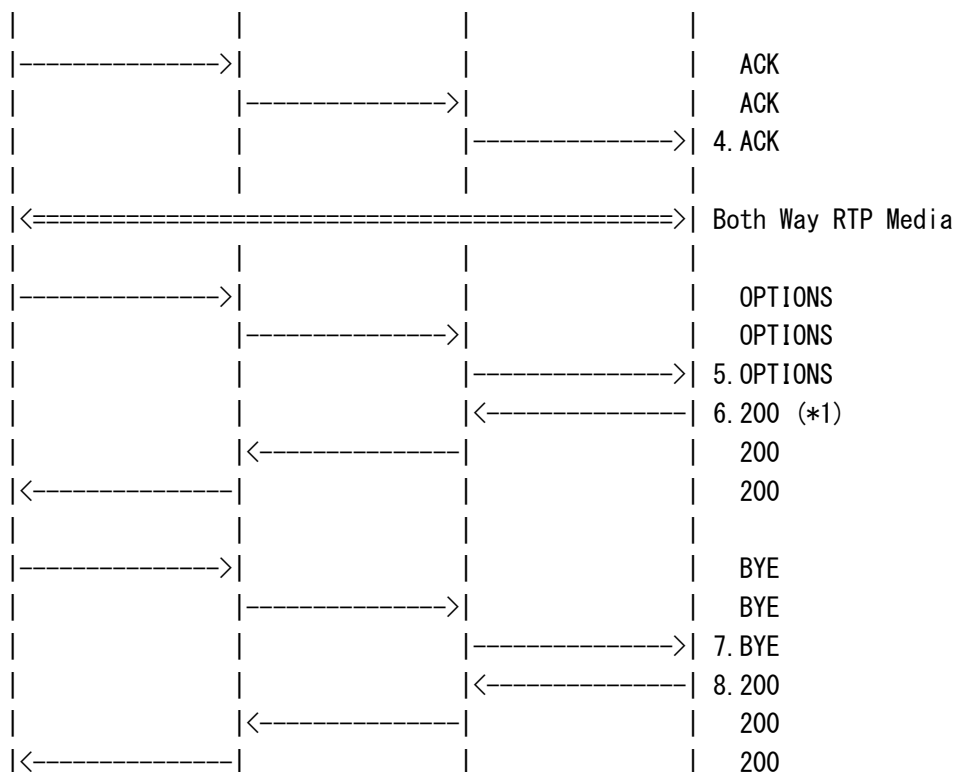
### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send OPTIONS.
6. Receive 200 OK. (\*1)
7. Send BYE.
8. Receive 200 OK.

#### === Message example ===

##### 5.OPTIONS Proxy -> NUT

```

OPTIONS sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
    
```



To: NUT <sip:NUT@under.test.com> ;tag=9fxced76xx  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 4 OPTIONS  
Contact: <sip:UA1@client.atlanta.example.com>  
Accept: application/sdp  
Content-Length: 0

- \* From tag is equal to 1.INVITE's.
- \* To tag is equal to 3.200's.
- \* CSeq number is incremented from 4.ACK's.

#### **[OBSERVABLE RESULTS]**

\*1:200 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "200". [RFC3261-11-1,3]

- Header fields:  
See generic\_response

- inside of a dialog

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq-2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\* Allow  
Should exist. [RFC3261-11-4]



- \* Accept  
Should exist. [RFC3261-11-4]

- \* Accept-Encoding  
Should exist. [RFC3261-11-4]

- \* Accept-Language  
Should exist. [RFC3261-11-4]

- \* Supported  
Should exist. [RFC3261-11-4]

- Bodies:  
See generic\_SDP

Should exist. [RFC3261-11-6]

## [REFERENCE]

[RFC3261-11-4, 5]

### 11 Querying for Capabilities

The SIP method OPTIONS allows a UA to query another UA or a proxy server as to its capabilities. This allows a client to discover information about the supported methods, content types, extensions, codecs, etc. without "ringing" the other party. For example, before a client inserts a Require header field into an INVITE listing an option that it is not certain the destination UAS supports, the client can query the destination UAS with an OPTIONS to see if this option is returned in a Supported header field. All UAs MUST support the OPTIONS method.

#### 11.2 Processing of OPTIONS Request

Allow, Accept, Accept-Encoding, Accept-Language, and Supported header field fields SHOULD be present in a 200 (OK) response to an OPTIONS request. If the response is generated by a proxy, the Allow header field field SHOULD be omitted as it is ambiguous since a proxy is method agnostic. Contact header fields MAY be present in a 200 (OK) response and have the same semantics as in a 3xx response. That is, they may list a set of alternative names and methods of reaching the user. A Warning header field MAY be present.



### 4.11.3 UA-12-2-1 - Receipt of OPTIONS when a UAS is busy over a dialog

#### [NAME]

UA-12-2-1 – Receipt of OPTIONS when the UAS is busy over a dialog.

#### [PURPOSE]

Verify that a NUT properly receives an OPTIONS request when the NUT is busy, that is, when INVITE is in a different dialog from the dialog where OPTIONS request is.

#### [REQUIREMENT]

Only when a NUT supports the function of OPTIONS request.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

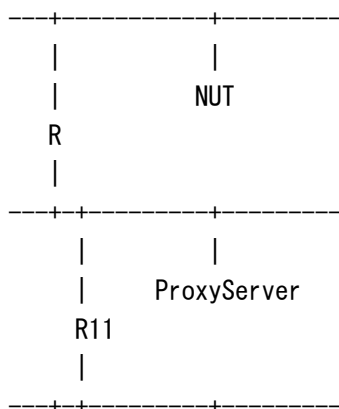
#### [PARAMETER]

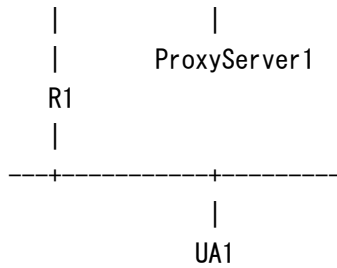
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

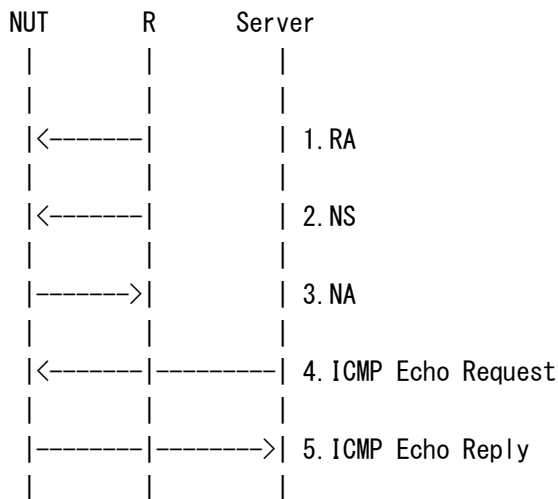




#### [CONFIGURATION for NUT]

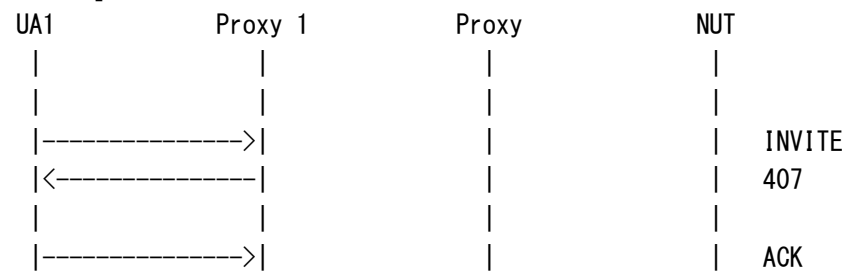
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

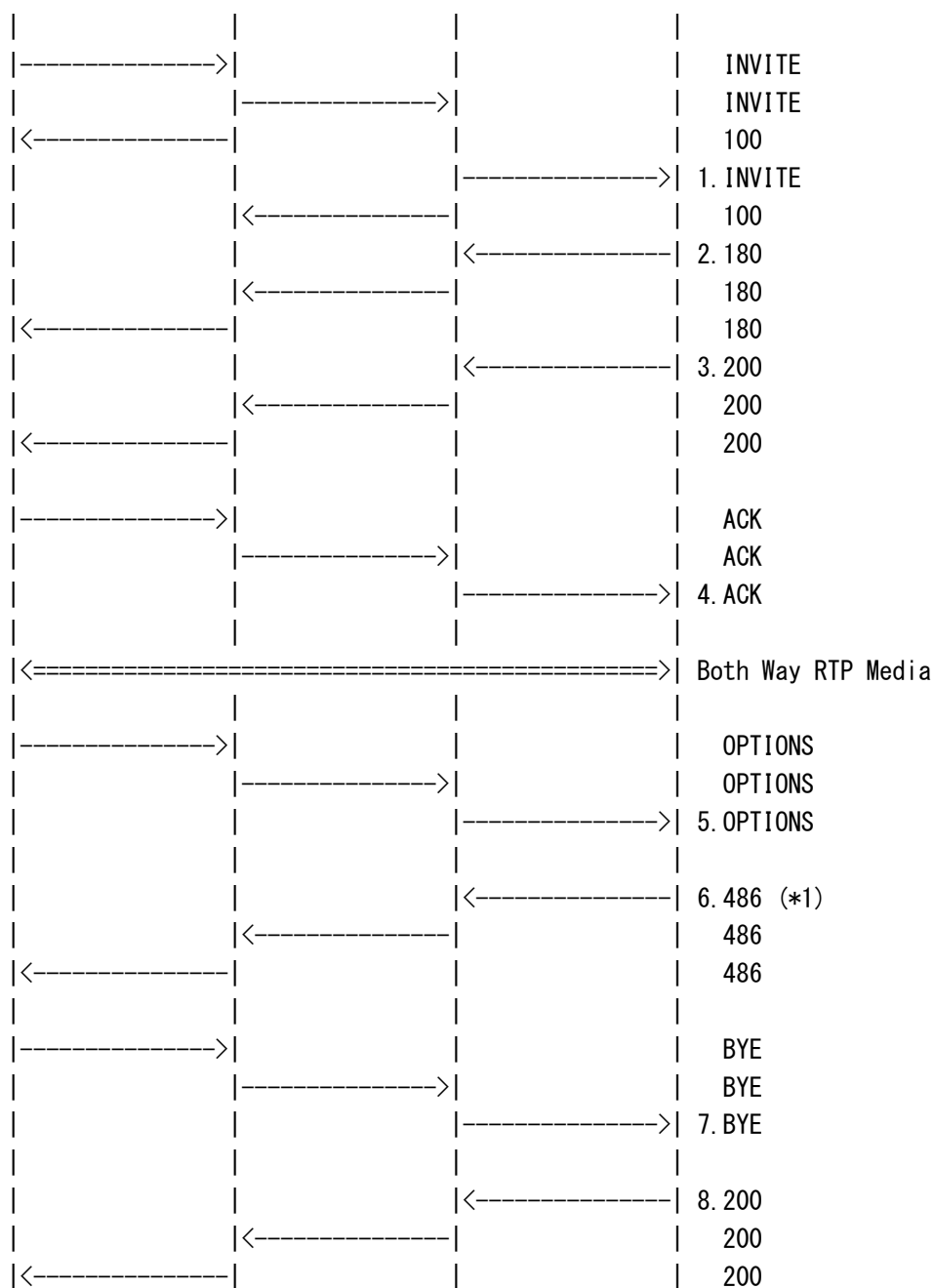
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ACK.
5. Send OPTIONS.
6. Receive 486 Busy Here. (\*1)
7. Send BYE.
8. Receive 200 OK.



=== Message example ===

#### 5.OPTIONS Proxy -> NUT

```
OPTIONS sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
    ;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
    <sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 OPTIONS
Contact: <sip:UA1@client.atlanta.example.com>
Accept: application/sdp
Content-Length: 0
```

- \* From tag is different from 1.INVITE's.
- \* To tag does not exist.
- \* Call-ID is different from 1.INVITE's.

#### [OBSERVABLE RESULTS]

\* 1:486 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "486". [RFC3261-11-1,3]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter



contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

- Bodies:  
See generic\_SDP

## [REFERENCE]

[RFC3261-11-3]

### 11.2 Processing of OPTIONS Request

The response to an OPTIONS is constructed using the standard rules for a SIP response as discussed in Section 8.2.6. The response code chosen MUST be the same that would have been chosen had the request been an INVITE. That is, a 200 (OK) would be returned if the UAS is ready to accept a call, a 486 (Busy Here) would be returned if the UAS is busy, etc. This allows an OPTIONS request to be used to determine the basic state of a UAS, which can be an indication of whether the UAS will accept an INVITE request.

## 4.12 DNS

### 4.12.1 UA-13-2-1 - Successful Session with Proxy Failure (Caller)

#### [NAME]

UA-13-2-1 - Successful Session with Proxy Failure (Caller)

#### [PURPOSE]

Verify that a NUT (caller) properly processes a successful session with Proxy Failure.

#### [REQUIREMENT]

Only when a NUT supports the function of DNS or can configure an alternate server.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

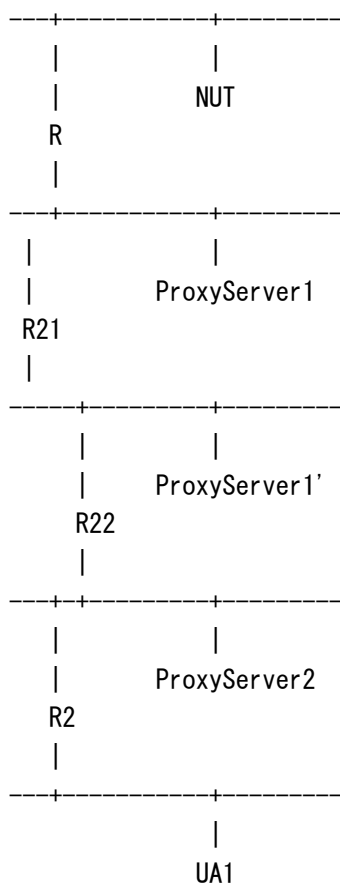
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server1	sip:ss.under.test.com:lr
ProxyServer1'	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com

ProxyServer2	sip:ss1.atlanta.example.com;lr
--------------	--------------------------------

#### [ADDRESS]

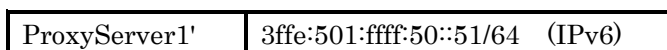
NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:50::50/64
ProxyServer1'(IPv6)	3ffe:501:ffff:50::51/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer2(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]



#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer1	sip:ss.under.test.com;lr
ProxyServer1'	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer1	3ffe:501:ffff:50::50/64 (IPv6)

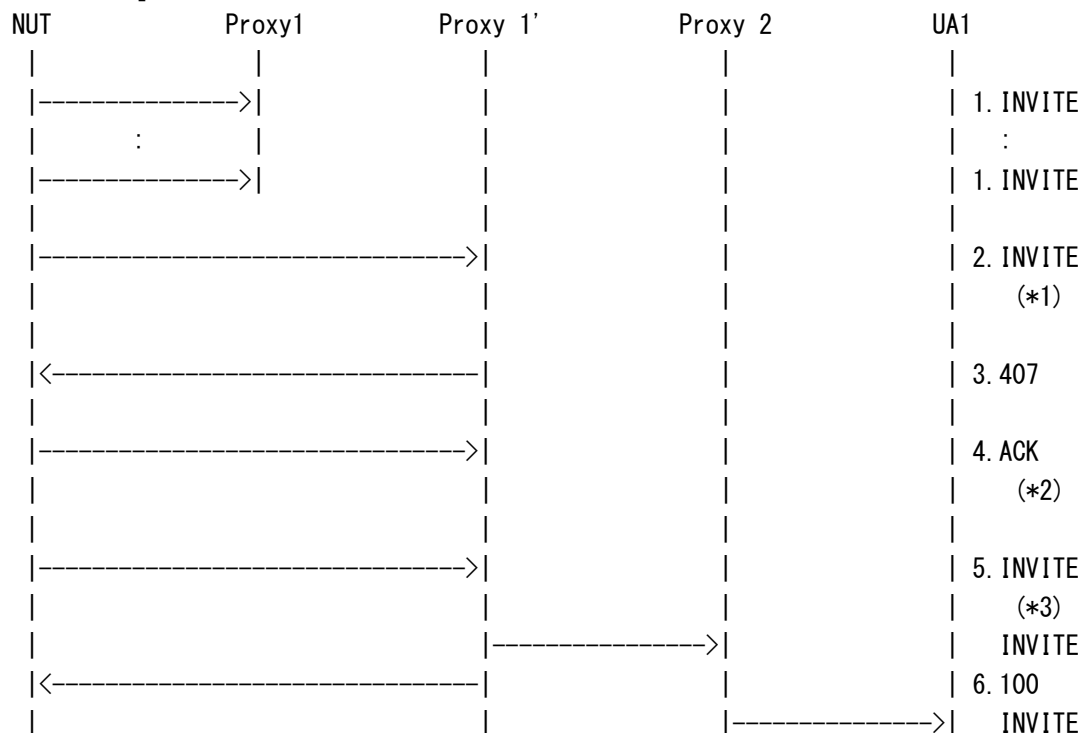


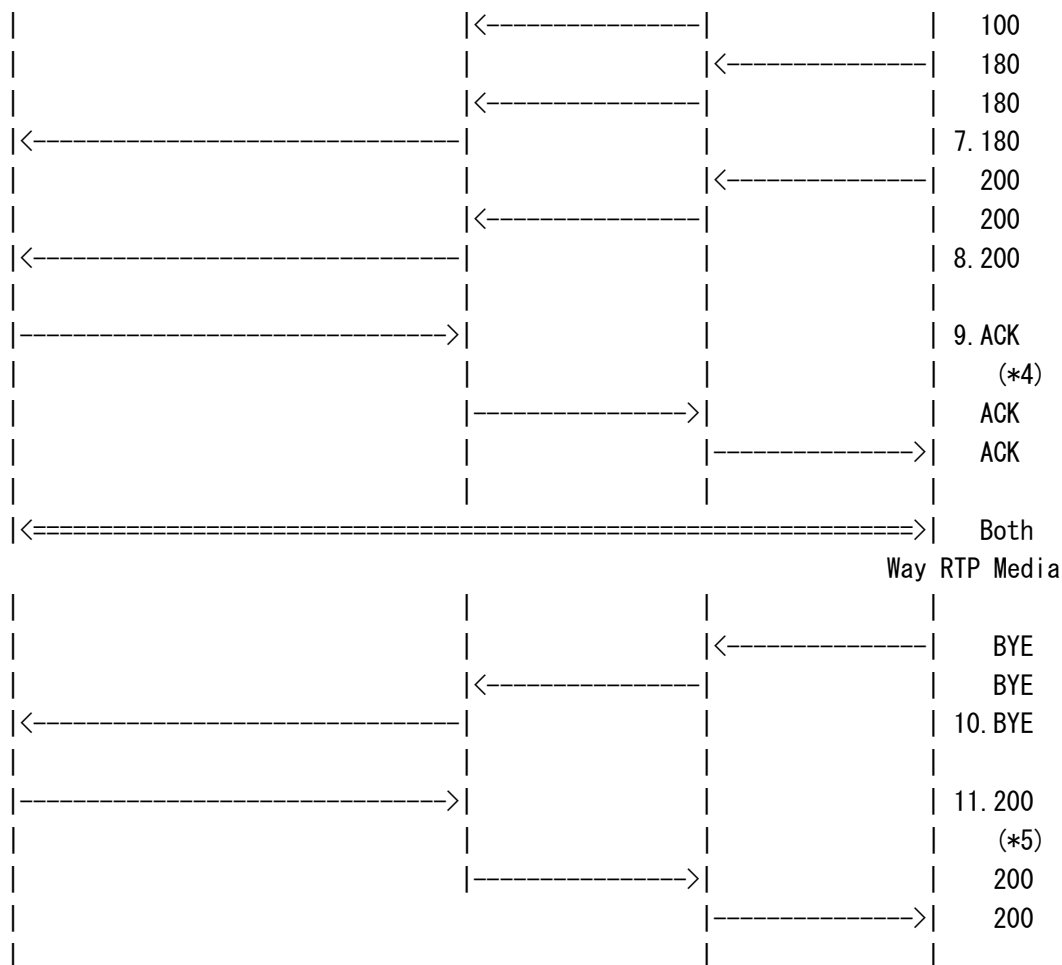
```

sequenceDiagram
    participant NUT
    participant R
    participant Server

    Note over NUT, R: 1. NUT sends RARP request to R
    Note over R: RARP request received
    Note over R, Server: R sends ARP request to Server
    Note over Server: ARP request received
    Note over Server: Server sends ARP reply to R
    Note over R: ARP reply received
    Note over R, NUT: R sends ICMP Echo Request to NUT
    Note over NUT: ICMP Echo Request received
    Note over NUT, Server: NUT sends ICMP Echo Reply to Server
    Note over Server: ICMP Echo Reply received
    
```

- [PROCEDURE]**





1. Receive INVITE.
2. Receive INVITE. (\*1)
3. Send 407 Proxy Authorization Required.
4. Receive ACK. (\*2)
5. Receive INVITE. (\*3)
6. Send 100 Trying.
7. Send 180 Ringing.
8. Send 200 OK.
9. Receive ACK. (\*4)
10. Send BYE.
11. Receive 200 OK. (\*5)

#### === Message example ===

##### 1. INVITE NUT -> Proxy 1

INVITE sip:UA1@atlanta.example.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK465b6d





Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## **2. INVITE NUT -> Proxy 1'**

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b8a  
Max-Forwards: 70  
Route: <sip:ss1.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 4Fde34wkd11wsGFds3@under.test.com  
CSeq: 1 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## **3. 407 Proxy Authorization Required Proxy 1' -> NUT**

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b8a



;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxcde76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=2421452  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="under.test.com", qop="auth",  
nonce="1ae6cbe5ea9c8e8df84fqnlc434a359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

#### 4. ACK NUT -> Proxy 1'

ACK sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b8a  
Route: <sip:ss1.under.test.com;lr>  
Max-Forwards: 70  
From: NUT <sip:NUT@under.test.com>;tag=9fxcde76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=2421452  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 1 ACK  
Content-Length: 0

#### 5. INVITE NUT -> Proxy 1'

INVITE sip:UA1@atlanta.example.com SIP/2.0  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
Max-Forwards: 70  
Route: <sip:ss1.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxcde76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:NUT@node.under.test.com>  
Proxy-Authorization: Digest username="NUT",  
realm="under.test.com",  
nonce="1ae6cbe5ea9c8e8df84fqnlc434a359", opaque="",  
qop=auth, nc=00000004, cnonce="6f54a149",  
uri="sip:UA1@atlanta.example.com",  
response="b51e504e73af54829e4f2bd7f8dc4654"  
Content-Type: application/sdp  
Content-Length: 151

v=0



o=NUT 2890844526 2890844526 IN IP6 3ffe:501:ffff:5::X  
s=-  
c=IN IP6 3ffe:501:ffff:5::X  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

## 6. 100 Trying Proxy 1' -> NUT

SIP/2.0 100 Trying  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 2 INVITE  
Content-Length: 0

## 7. 180 Ringing Proxy 1' -> NUT

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com:lr>,  
<sip:ss1.under.test.com:lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Length: 0

## 8. 200 OK Proxy 1' -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com:lr>,  
<sip:ss1.under.test.com:lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 2 INVITE



Contact: <sip:UA1@client.atlanta.example.com>

Content-Type: application/sdp

Content-Length: 147

v=0

o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1

s=-

c=IN IP6 3ffe:501:ffff:1::1

t=0 0

m=audio 3456 RTP/AVP 0

a=rtpmap:0 PCMU/8000

## 9. ACK NUT -> Proxy 1'

ACK sip:UA1@client.atlanta.example.com SIP/2.0

Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74b8g

Max-Forwards: 70

Proxy-Authorization: Digest username="NUT",

realm="under.test.com",

nonce="1ae6cbe5ea9c8e8df84fqnllec434a359", opaque="",

qop=auth, nc=00000004, cnonce="6f54a149",

uri="sip:UA1@atlanta.example.com",

response="b51e504e73af54829e4f2bd7f8dc4654"

Route: <sip:ss1.under.test.com:lr>,

<sip:ss1.atlanta.example.com:lr>

From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

To: UA1 <sip:UA1@atlanta.example.com>;tag=314159

Call-ID: 4Fde34wkd11wsGFDs3@under.test.com

CSeq: 2 ACK

Content-Length: 0

## 10. BYE Proxy 1' -> NUT

BYE sip:NUT@node.under.test.com SIP/2.0

Via: SIP/2.0/UDP ss1.under.test.com:5060;branch=z9hG4bK2d4790.1

Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1

;received=3ffe:501:ffff:20::20

Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7

;received=3ffe:501:ffff:1::1

Max-Forwards: 68

From: UA1 <sip:UA1@atlanta.example.com>;tag=314159

To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl

Call-ID: 4Fde34wkd11wsGFDs3@under.test.com



CSeq: 1 BYE  
Content-Length: 0

#### 11. 200 OK NUT -> Proxy 1'

SIP/2.0 200 OK  
Via: SIP/2.0/UDP ss1.under.test.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:50::51  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK721e418c4.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7  
;received=3ffe:501:ffff:1::1  
From: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
To: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
Call-ID: 4Fde34wkd11wsGFDs3@under.test.com  
CSeq: 1 BYE  
Content-Length: 0

#### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.(Proxy 1')

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE

- Header fields:  
See generic\_request

- outside of a dialog  
See generic\_Initial-INVITE

- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

\*2:ACK request from NUT.(Proxy 1')



As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK  
See generic\_non2xx-ACK
- Header fields:  
See generic\_request
- outside of a dialog  
See generic\_ACK  
See generic\_non2xx-ACK
- Bodies:  
See generic\_ACK  
See generic\_non2xx-ACK

\*3:INVITE request from NUT.(Proxy 1')

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE
- Header fields:  
See generic\_request
- outside of a dialog  
See generic\_Initial-INVITE

\* Call-ID

callid: Should be the same as that of "1.INVITE" request, with case-sensitivity on. [RFC3261-8-62]



\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

See generic\_digest-auth

- Bodies:

See generic\_Initial-INVITE

See generic\_SDP

\*4:ACK request from NUT.(Proxy 1')

As a SIP Message,

See generic\_message

As a SIP request,

- Request-Line:

See generic\_request

See generic\_ACK

See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

Must be the same as that of \*3 Proxy-Authorization. [RFC3261-22-10]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

\*5:200 response from NUT.(Proxy 1')

As a SIP Message,

See generic\_message

As a SIP request,



- Status-Line:  
See generic\_response
  - \* Status-Code: Must be "200". [RFC3261.22.2.1]
- Header fields:  
See generic\_response
- inside of a dialog  
See generic\_response
- \* Via
  - via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]
  - via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

#### [REFERENCE]

Sequence in Section 3.4., RFC3665

### 4.12.2 UA-13-2-2 - Forwarding of INVITE to an alternative server upon receipt of 503

#### [NAME]

UA-13-2-2 - Forwarding of INVITE to an alternative server upon receipt of 503

#### [PURPOSE]

Verify that a NUT properly forwards an INVITE to an alternative server when receiving a 503 response.

#### [REQUIREMENT]

Only when a NUT supports the function of DNS or can configure an alternate server.

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

#### [PARAMETER]

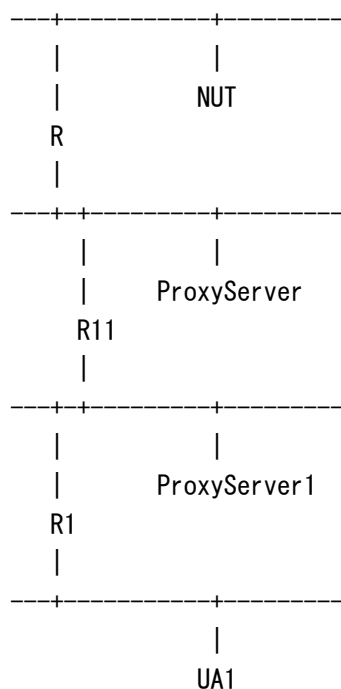
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr



# [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
ProxyServer'(IPv6)	3ffe:501:ffff:50::51/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

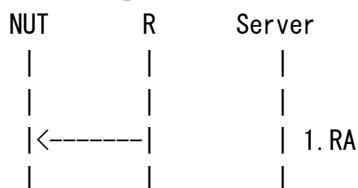
# [TOPOLOGY]

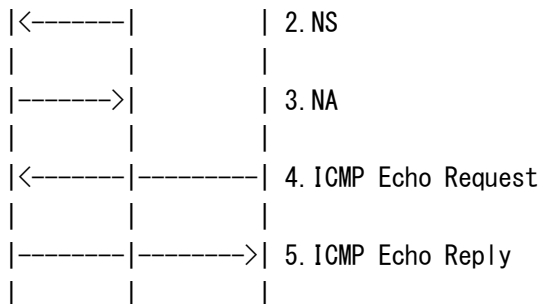


# [CONFIGURATION for NUT]

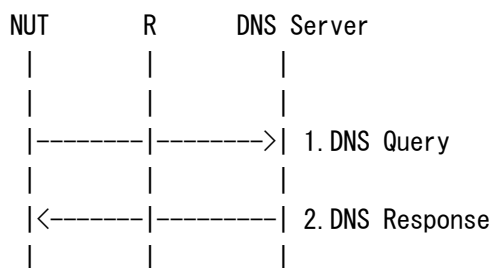
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

# [INITIALIZATION]





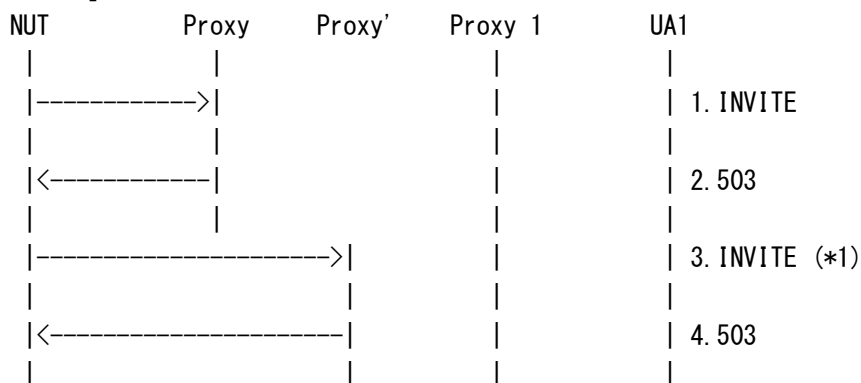
1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.



ss.under.test.com resolves to two IP addresses.

(ss.under.test.com	3ffe:501:ffff:50::50)
(ss.under.test.com	3ffe:501:ffff:50::51)

#### [PROCEDURE]



1. Receive INVITE.
2. Send 503 Service Unavailable.
3. Receive INVITE. (\*1)
4. Send 503 Service Unavailable.



=== Message example ===

**2. 503 Service Unavailable Proxy -> NUT**

SIP/2.0 503 Service Unavailable  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bKbf9f45  
;received=3ffe:501:ffff:5::X  
From: NUT <sip:NUT@under.test.com>;tag=9fxcde76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=53fHlqlQ3  
Call-ID: 2xTb9vxSit55XU7p8@under.test.com  
Retry-After: 3600  
CSeq: 3 INVITE  
Content-Length: 0

**[OBSERVABLE RESULTS]**

\*1:INVITE request from NUT.(Proxy')

As a SIP Message,  
See generic\_message  
The destination address should be equal to 3ffe:501:ffff:50::51.  
[RFC3261-21-27]

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_Initial-INVITE
- Header fields:  
See generic\_request
- outside of a dialog  
See generic\_Initial-INVITE
- Bodies:  
See generic\_Initial-INVITE  
See generic\_SDP

**[REFERENCE]**

[RFC3261-21-27, 28]

21.5.4 503 Service Unavailable

A client (proxy or UAC) receiving a 503 (Service Unavailable) SHOULD attempt to forward the request to an alternate server. It SHOULD NOT



forward any other requests to that server for the duration specified in the Retry-After header field, if present.

## 4.13 Transport

### 4.13.1 UA-14-2-1 - Receipt of INVITE with additional bytes in the transport packet

#### [NAME]

UA-14-2-1 - Receipt of INVITE with additional bytes in the transport packet

#### [PURPOSE]

Verify that a NUT properly processes when receiving an INVITE request containing additional bytes in the transport packet.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

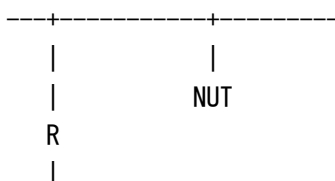
#### [PARAMETER]

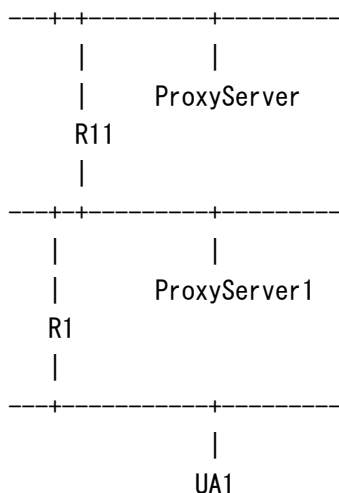
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

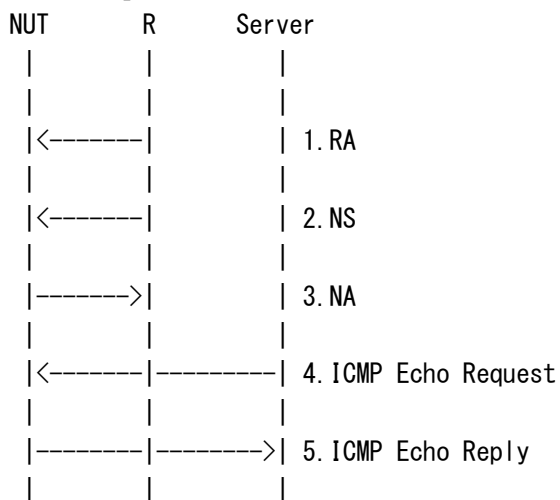




#### [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

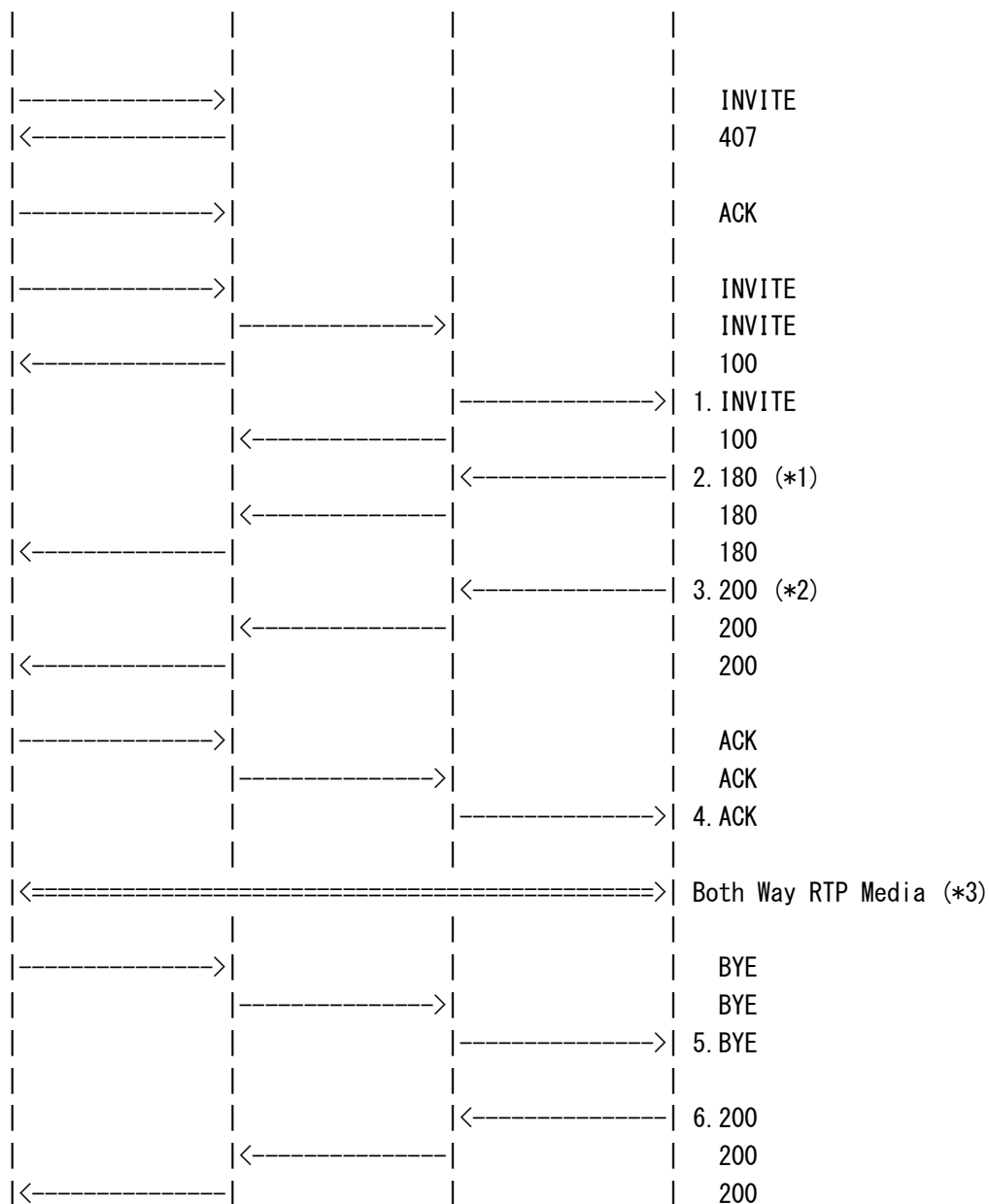
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]

UA1                      Proxy 1                      Proxy                      NUT



1. Send INVITE.
2. Receive 180 Ringing. (\*1)
3. Receive 200 OK. (\*2)
4. Send ACK.
- (\*3)
5. Send BYE.
6. Receive 200 OK.

=== Message example ===

1. INVITE Proxy -> NUT



INVITE sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
Record-Route: <sip:ss.under.test.com;lr>,  
<sip:ss1.atlanta.example.com;lr>  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 0  
  
v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
  
\* Content-Length: 0

#### **4.ACK Proxy -> NUT**

ACK sip:NUT@node.under.test.com SIP/2.0  
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1  
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
;received=3ffe:501:ffff:20::20  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:1::1  
Max-Forwards: 68  
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl  
To: NUT <sip:NUT@under.test.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 2 ACK  
Content-Length: 151



v=0  
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 49170 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

\* Answer is included.

The port number in the m= line is different from 1.INVITE's.

### **[OBSERVABLE RESULTS]**

\*1:180 response from NUT. [Optional]

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "1xx". [RFC3261 8.2.6.1]

- Header fields:  
See generic\_response

- outside of a dialog

\* Via

via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]

via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route

Must exist. [ORq -2]

Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]

rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

\*2:200 response from NUT.





As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
See generic\_200-for-INVITE  
Status-Code: Must be "200". [RFC3261 12.1, 13, 21.2]

- Header fields:  
See generic\_response

- outside of a dialog  
See generic\_200-for-INVITE

\* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

\* Record-Route  
Must exist. [ORq -2]  
Must copy all Record-Route header field values from the request into the response. [RFC3261-12-2, 9]  
rec-route: Must maintain the order of Record-Route header field values. [RFC3261-12-3]

- Bodies:  
See generic\_200-for-INVITE  
See generic\_SDP

The Offer Must be included. [RFC3261-18-38, RFC3261-13-10]

\*2:Both Way RTP Media established.

Must send RTP packets on the port specified in the "4.ACK".

#### **[REFERENCE]**

[RFC3261-18-38, 39, 40]  
18.3 Framing

In the case of message-oriented transports (such as UDP), if the message has a Content-Length header field, the message body is assumed to contain that many bytes. If there are additional bytes in the transport packet beyond the end of the body, they MUST be discarded. If the transport packet ends before the end of the message body, this is considered an error. If the message is a response, it MUST be discarded. If the message is a request, the element SHOULD generate a 400 (Bad Request) response. If the message has no Content-Length header field, the message body is assumed to end at the end of the transport packet.

#### 4.13.2 UA-14-2-2 - Transport packet of a response ending before the end of the message body

##### [NAME]

UA-14-2-2 - Transport packet of response ending before the end of the message body

##### [PURPOSE]

Verify that a NUT properly ignores when a transport packet of a response that ends before the message body ends.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

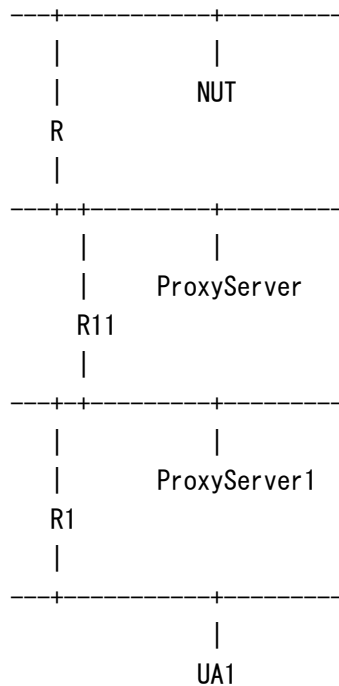
##### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

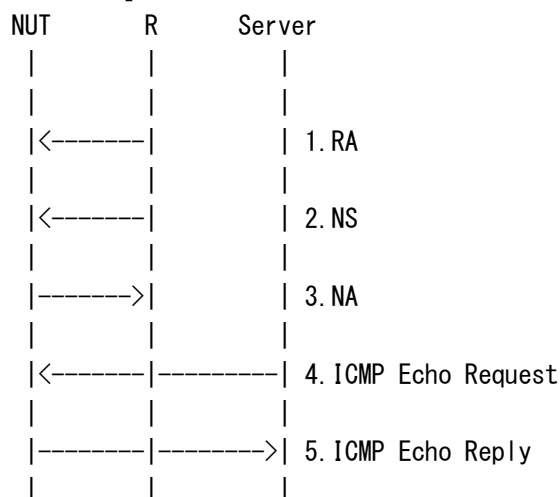
## [TOPOLOGY]



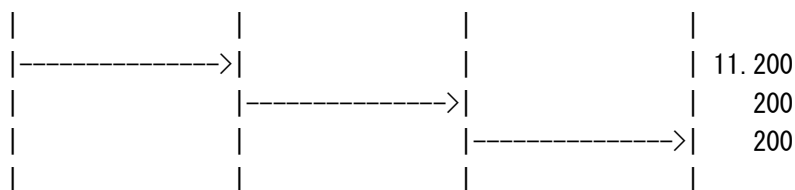
## [CONFIGURATION for NUT]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

## [INITIALIZATION]







1. Receive INVITE.
2. Send 407 Proxy Authorization Required.
3. Receive ACK.
4. Receive INVITE.
5. Send 100 Trying.
6. Send 180 Ringing.
7. Send 200 OK.
- (\*1)
8. Send 200 OK.
9. Receive ACK. (\*2)
10. Send BYE.
11. Receive 200 OK.

#### === Message example ===

##### 7. 200 OK Proxy -> NUT

```

SIP/2.0 200 OK
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:5::X
Record-Route: <sip:ss1.atlanta.example.com:lr>,
<sip:ss.under.test.com:lr>
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159
Call-ID: 3848276298220188511@under.test.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 350

```

```

v=0
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000

```



- \* Larger Content-Length value.  
Actual body size is equal to 147 bytes.

## 8. 200 OK Proxy -> NUT

SIP/2.0 200 OK  
Via: SIP/2.0/UDP node.under.test.com:5060;branch=z9hG4bK74bf9  
;received=3ffe:501:ffff:5::X  
Record-Route: <sip:ss1.atlanta.example.com;lr>,  
<sip:ss.under.test.com;lr>  
From: NUT <sip:NUT@under.test.com>;tag=9fxced76sl  
To: UA1 <sip:UA1@atlanta.example.com>;tag=314159  
Call-ID: 3848276298220188511@under.test.com  
CSeq: 2 INVITE  
Contact: <sip:UA1@client.atlanta.example.com>  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UA1 2890844527 2890844527 IN IP6 3ffe:501:ffff:1::1  
s=-  
c=IN IP6 3ffe:501:ffff:1::1  
t=0 0  
m=audio 3456 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

### [OBSERVABLE RESULTS]

- \*1:after 200 response from Proxy.

Must not send ACK and RTP packets, because Content-Length value is larger than size of body. [RFC3261-18-39]

- \*2:ACK request from NUT.

As a SIP Message,  
See generic\_message

As a SIP request,

- Request-Line:  
See generic\_request  
See generic\_ACK



See generic\_2xx-ACK

- Header fields:

See generic\_request

- inside of a dialog

See generic\_ACK

See generic\_2xx-ACK

\* Proxy-Authorization

Must exist. [RFC3261.22.3],[RFC3261-22-22]

Must be the same as that of Proxy-Authorization in "4.INVITE".  
[RFC3261-13-22]

\* Route

Must exist. [ORq-2]

Must contain the Record-Route values of "7.200 OK" in reverse order, including all parameters, and the first URI in the route set contains the lr parameter. [RFC3261-12-23, 48]

- Bodies:

See generic\_ACK

See generic\_2xx-ACK

## [REFERENCE]

[RFC3261-18-38, 39, 40]

### 18.3 Framing

In the case of message-oriented transports (such as UDP), if the message has a Content-Length header field, the message body is assumed to contain that many bytes. If there are additional bytes in the transport packet beyond the end of the body, they **MUST** be discarded. If the transport packet ends before the end of the message body, this is considered an error. If the message is a response, it **MUST** be discarded. If the message is a request, the element **SHOULD** generate a 400 (Bad Request) response. If the message has no Content-Length header field, the message body is assumed to end at the end of the transport packet.

## 4.13.3 UA-14-2-3 - Transport packet of request ending before the end of the message body

[NAME]

UA-14-2-3 - Transport packet of request ending before the end of the message body

#### [PURPOSE]

Verify that a NUT properly sends a 400 response when a transport packet of a request that ends before the message body ends.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

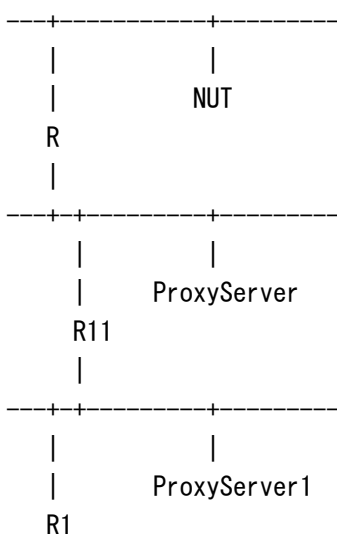
#### [PARAMETER]

NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

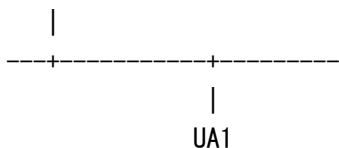
#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]



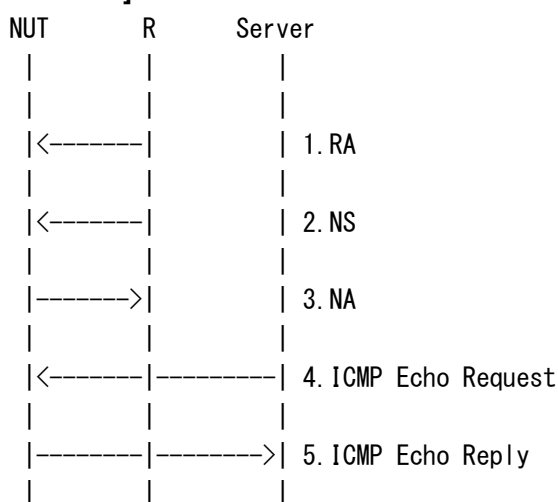




# [CONFIGURATION for NUT]

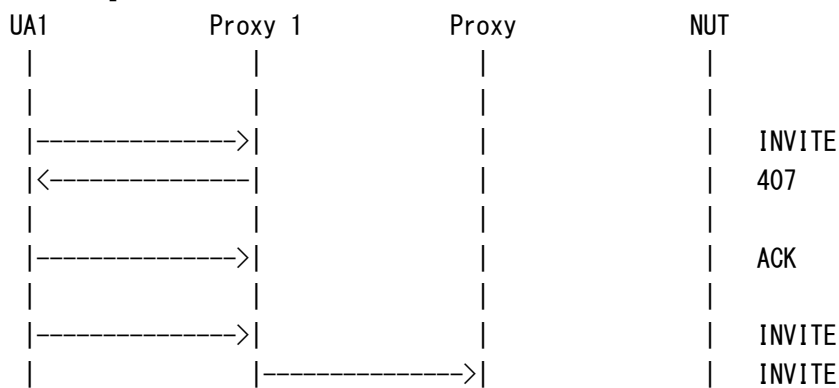
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

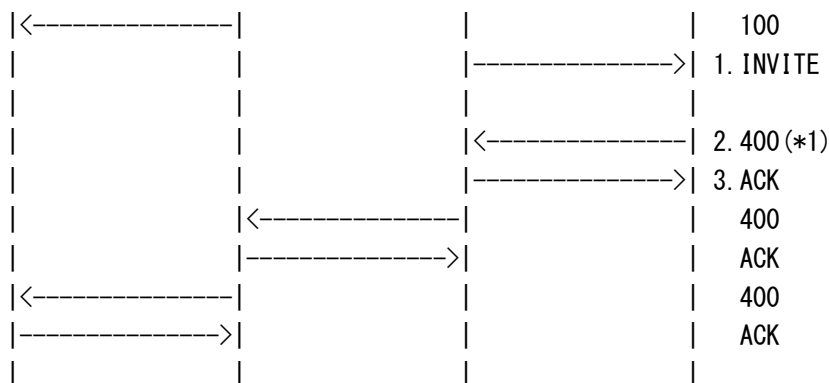
# [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

# [PROCEDURE]





1. Send INVITE.
2. Receive 400 Bad Request. (\*1)
3. Send ACK.

### === Message example ===

#### 1. INVITE Proxy -> NUT

```

INVITE sip:NUT@node.under.test.com SIP/2.0
Via: SIP/2.0/UDP ss.under.test.com:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/UDP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=3ffe:501:ffff:20::20
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=3ffe:501:ffff:1::1
Max-Forwards: 68
Record-Route: <sip:ss.under.test.com;lr>,
<sip:ss1.atlanta.example.com;lr>
From: UA1 <sip:UA1@atlanta.example.com>;tag=9fxced76sl
To: NUT <sip:NUT@under.test.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:UA1@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 350
  
```

```

v=0
o=UA1 2890844526 2890844526 IN IP6 3ffe:501:ffff:1::1
s=-
c=IN IP6 3ffe:501:ffff:1::1
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
  
```



- \* Larger Content-Length value.  
Actual body size is equal to 148 bytes.

## **[OBSERVABLE RESULTS]**

\* 1:400 response from NUT.

As a SIP Message,  
See generic\_message

As a SIP response,

- Status-Line:  
See generic\_response  
Status-Code: Must be "400", because Content-Length value is larger than size of body. [RFC3261-18-40]
- Header fields:  
See generic\_response
  - \* Via  
via-received: Must be added if the host portion of the "sent-by" parameter contains a domain name. [RFC3261-18-27]  
via-received: Must contain the source address from which the packet was received. [RFC3261-18-28]

## **[REFERENCE]**

[RFC3261-18-38, 39, 40]

### 18.3 Framing

In the case of message-oriented transports (such as UDP), if the message has a Content-Length header field, the message body is assumed to contain that many bytes. If there are additional bytes in the transport packet beyond the end of the body, they **MUST** be discarded. If the transport packet ends before the end of the message body, this is considered an error. If the message is a response, it **MUST** be discarded. If the message is a request, the element **SHOULD** generate a 400 (Bad Request) response. If the message has no Content-Length header field, the message body is assumed to end at the end of the transport packet.

## 4.14 ICMP

### 4.14.1 UA-15-2-1 - Receipt of “ICMP destination unreachable” for a previously sent request

#### [NAME]

UA-15-2-1 - Receipt of “ICMP destination unreachable” for a previously sent request

#### [PURPOSE]

Verify that a NUT properly doesn’t send any request when receiving an “ICMP destination unreachable” error message in a sent request.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

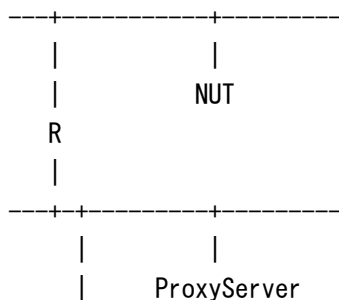
#### [PARAMETER]

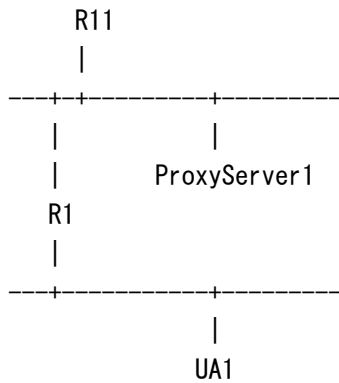
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

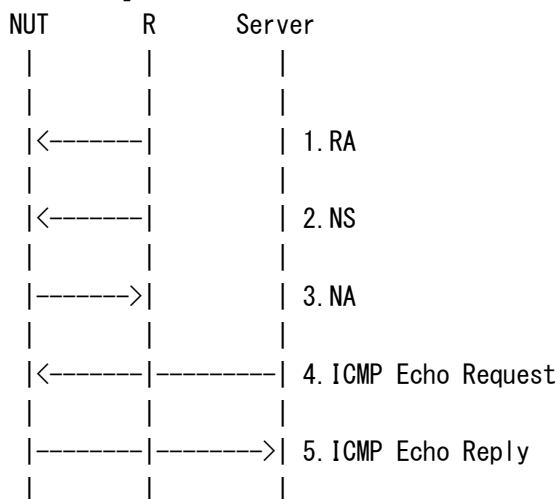




#### [CONFIGURATION for NUT]

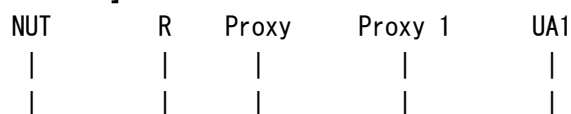
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

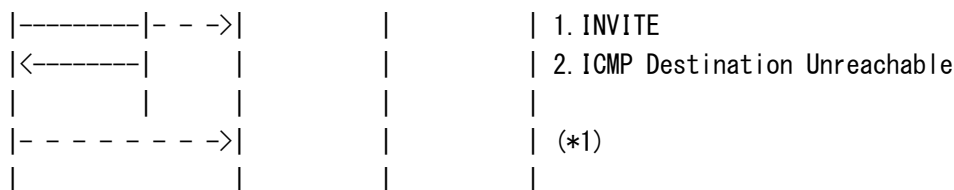
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Receive INVITE.
  2. Send ICMP Destination Unreachable.
- (\*1)

### [OBSERVABLE RESULTS]

\*1: after ICMP Destination Unreachable

Must not send the same INVITE, which has the same Via-branch parameter.  
[RFC3261-8-42, RFC3261-18-42]

( If UA wants, resending of INVITE is OK. But the INVITE must not have the same Via-branch parameter.)

### [REFERENCE]

[RFC3261-8-41, 42]

#### 8.1.3.1 Transaction Layer Errors

In some cases, the response returned by the transaction layer will not be a SIP message, but rather a transaction layer error. When a timeout error is received from the transaction layer, it **MUST** be treated as if a 408 (Request Timeout) status code has been received. If a fatal transport error is reported by the transport layer (generally, due to fatal ICMP errors in UDP or connection failures in TCP), the condition **MUST** be treated as a 503 (Service Unavailable) status code.

[RFC3261-18-42]

#### 18.4 Error Handling

If the transport user asks for a message to be sent over an unreliable transport, and the result is an ICMP error, the behavior depends on the type of ICMP error. Host, network, port or protocol unreachable errors, or parameter problem errors **SHOULD** cause the transport layer to inform the transport user of a failure in sending. Source quench and TTL exceeded ICMP errors **SHOULD** be ignored.

#### 4.14.2 UA-15-2-2 - Receipt of “ICMP time exceeded” for a previously sent request

##### [NAME]

UA-15-2-2 - Receipt of “ICMP time exceeded” for a previously sent request

##### [PURPOSE]

Verify that a NUT properly retransmits a request after T1 seconds when receiving an “ICMP time exceeded” error message for a sent request.

##### [REQUIREMENT]

NONE

##### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

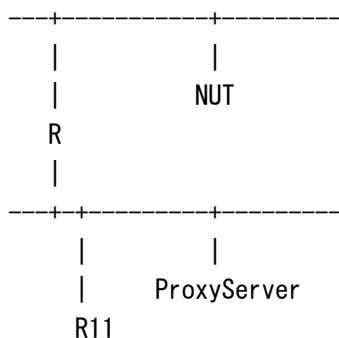
##### [PARAMETER]

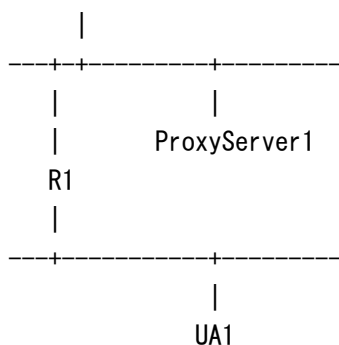
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr
T1	0.5sec

##### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

##### [TOPOLOGY]

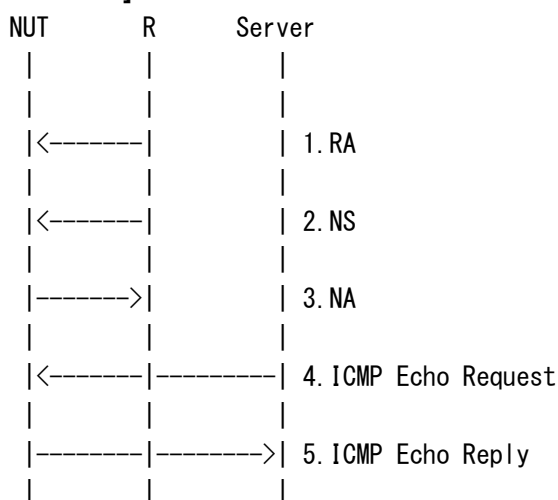




### [CONFIGURATION for NUT]

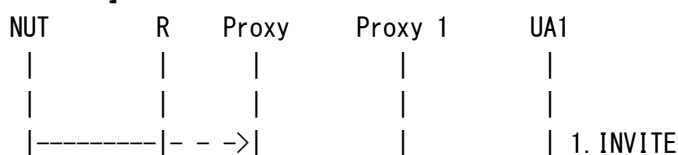
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com:lr
UA1(AOR)	sip:UA1@atlanta.example.com
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

### [INITIALIZATION]

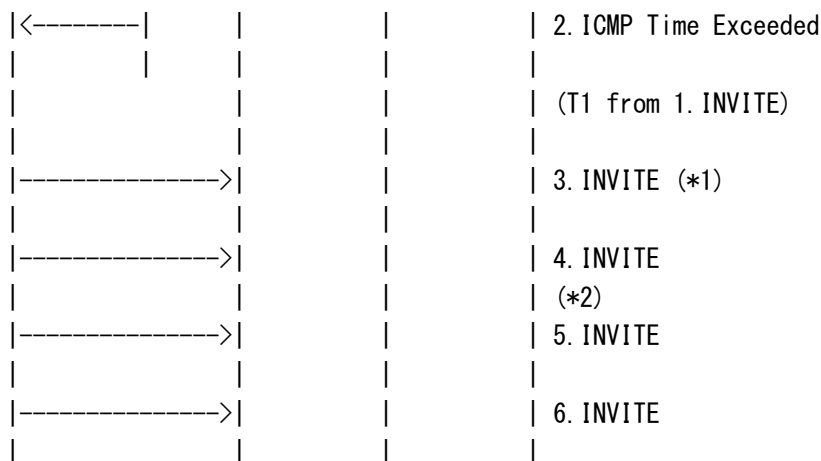


1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

### [PROCEDURE]







1. Receive INVITE.
2. Send ICMP Time Exceeded.
3. Receive INVITE. (\*1)
4. Receive INVITE.  
(\*2)
5. Receive INVITE.
6. Receive INVITE.

#### [OBSERVABLE RESULTS]

\*1:INVITE request from NUT.

Should be the same (retransmitted) INVITE. [RFC3261-18-43]

\*2 INVITE request from NUT.

Must be retransmitted with intervals that double after each transmission( $2 \times T1$ ).

[RFC3261-17-8][RFC3261-17-9][RFC3261-17-10][RFC3261-17-14][RFC3261-18-43]

#### [REFERENCE]

[RFC3261-18-42]

##### 18.4 Error Handling

If the transport user asks for a message to be sent over an unreliable transport, and the result is an ICMP error, the behavior depends on the type of ICMP error. Host, network, port or protocol unreachable errors, or parameter problem errors SHOULD cause the transport layer to inform the transport user of a failure in sending. Source quench and TTL exceeded ICMP errors SHOULD be ignored.

### 4.14.3 UA-15-2-3 - Receipt of “ICMP time exceeded” for a previously sent response

#### [NAME]

UA-15-2-3 - Receipt of “ICMP time exceeded” for a previously sent response

#### [PURPOSE]

Verify that a NUT properly retransmits a response after T1 seconds when receiving an “ICMP time exceeded” error message in a sent response.

#### [REQUIREMENT]

NONE

#### [TARGET]

SIP User Agent, SIP Endpoint, SIP Back-to-Back User Agent

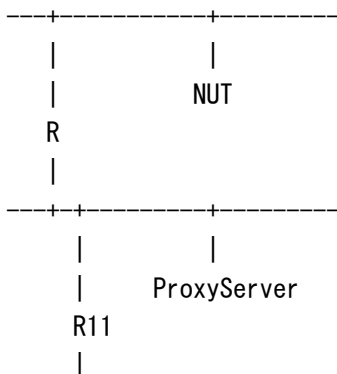
#### [PARAMETER]

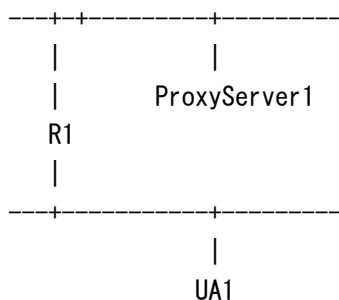
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
Proxy Server	sip:ss.under.test.com;lr
UA1(AOR)	sip:UA1@atlanta.example.com
UA1(Contact)	sip:UA1@client.atlanta.example.com
ProxyServer1	sip:ss1.atlanta.example.com;lr

#### [ADDRESS]

NUT(IPv6)	3ffe:501:ffff:5::X/64 (Interface ID=Auto Negotiation)
R(IPv6)	3ffe:501:ffff:5::1/64
ProxyServer(IPv6)	3ffe:501:ffff:50::50/64
UA1(IPv6)	3ffe:501:ffff:1::1/64
ProxyServer1(IPv6)	3ffe:501:ffff:20::20/64

#### [TOPOLOGY]

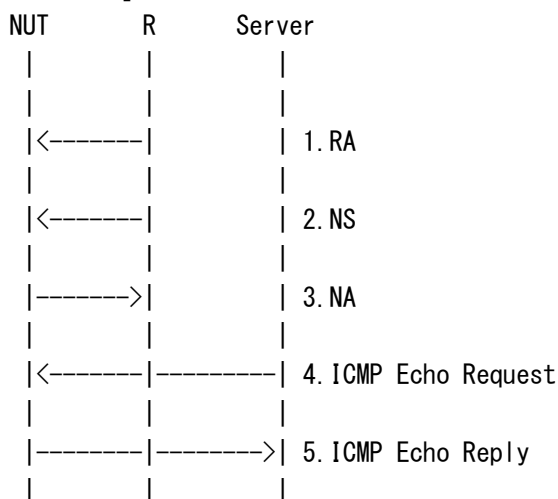




#### [CONFIGURATION for NUT]

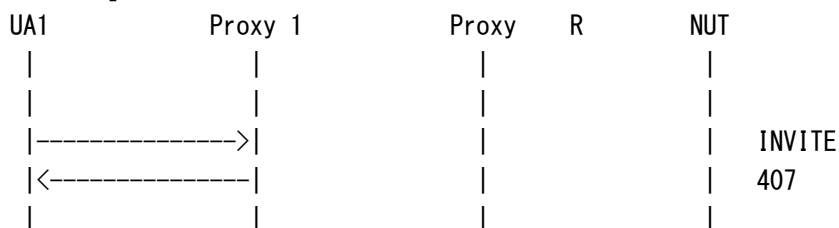
NUT(AOR)	sip:NUT@under.test.com
NUT(Contact)	sip:NUT@node.under.test.com
ProxyServer	sip:ss.under.test.com;lr
ProxyServer	3ffe:501:ffff:50::50/64 (IPv6)

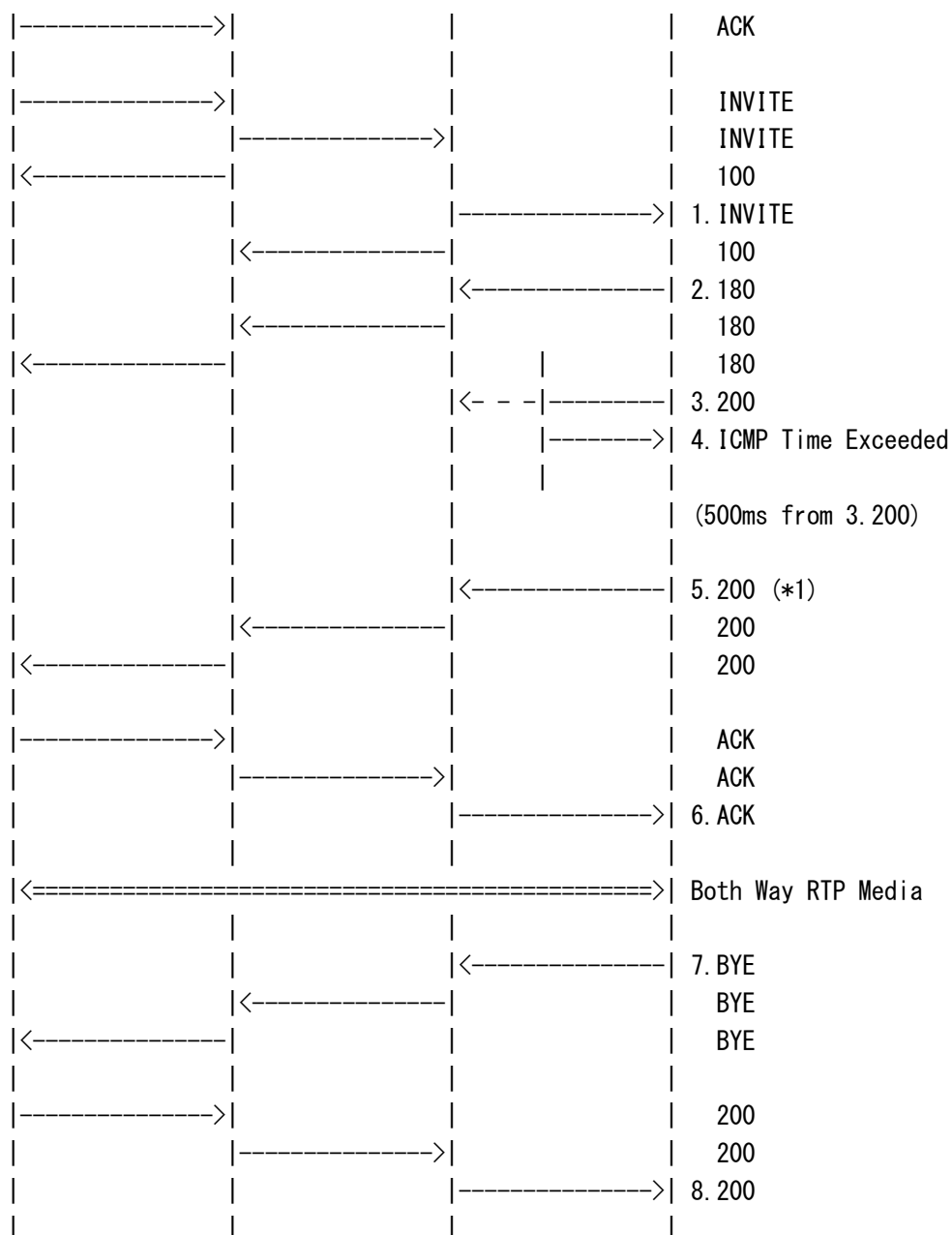
#### [INITIALIZATION]



1. Send Router Advertisement.
2. Send Neighbor Solicitation.
3. Receive Neighbor Advertisement.
4. Send ICMP Echo Request.
5. Receive ICMP Echo Reply.

#### [PROCEDURE]





1. Send INVITE.
2. Receive 180 Ringing.
3. Receive 200 OK.
4. Send ICMP Time Exceeded.
5. Receive 200 OK. (\*1)
6. Send ACK.
7. Receive BYE.
8. Send 200 OK.



### **[OBSERVABLE RESULTS]**

\*1:200 response from NUT.

Should be the same (retransmitted) 200 response. [RFC3261-18-43]

### **[REFERENCE]**

[RFC3261-18-42]

#### 18.4 Error Handling

If the transport user asks for a message to be sent over an unreliable transport, and the result is an ICMP error, the behavior depends on the type of ICMP error. Host, network, port or protocol unreachable errors, or parameter problem errors SHOULD cause the transport layer to inform the transport user of a failure in sending. Source quench and TTL exceeded ICMP errors SHOULD be ignored.



\*\*\*\*\*

**Copyright (C) 2005-2010 IPv6 Forum. All Rights Reserved.**

This original documentation is produced by SIP IPv6 SWG members of Certification WG in the IPv6 Promotion Council. The SWG members currently include Nippon Telegraph and Telephone Corporation (NTT), Yokogawa Electric Corporation, University of New Hampshire InterOperability Laboratory (UNH-IOL), and NTT Advanced Technology Corporation (NTT-AT).

No part of this documentation may be reproduced for any purpose without prior permission.



## AUTHORS' LIST

Hiroshi Miyata (Yokogawa Electric Corporation)

Yukiyo Akisada (Yokogawa Electric Corporation)

Timothy Winters (UNH-IOL)

James Swan (UNH-IOL)

Yoshio Yoshida (NTT-AT)

Kenzo Kodama (NTT-AT)

Yoshihiro Inoue (NTT-AT)

Naomi Orimo(NTT-AT)