Section	SubSection	Page	No	Description	Status
7		27	1	The start-line, each message-header line, and the empty line **MUST** be terminated by a carriage-return line-feed sequence (CRLF).	M
7		27	2	Note that the empty line **MUST** be present even if the message-body is not.	M
7	7.1	28	3	The Request-URI **MUST NOT** contain unescaped spaces or control characters and **MUST NOT** be enclosed in "<>".	M
			4		M
7	7.1	30	5	To be compliant with this specification, applications sending SIP messages **MUST** include a SIP-Version of "SIP/2.0".	M
7	7.1	30	6	The SIP-Version string is case-insensitive, but implementations **MUST** send upper-case.	M
7	7.3.1	30	7	However, it is **RECOMMENDED** that header fields which are needed for proxy processing (Via, Route, Record-Route, Proxy-Require, Max-Forwards, and Proxy-Authorization, for example) appear towards the top of the message to facilitate rapid parsing.	R
7	7.3.1	30	8	It **MUST** be possible to combine the multiple header field rows into one "field-name: field-value" pair, without changing the semantics of the message, by appending each subsequent field-value to the first, each separated by a comma.	M
7	7.3.1	31	9	The exceptions to this rule are the WWW-Authenticate, Authorization, Proxy-Authenticate, and Proxy-Authorization header fields. Multiple header field rows with these names MAY be present in a message, but since their grammar does not follow the general form listed in Section 7.3, they **MUST NOT** be combined into a single header field row.	M

Section	SubSection	Page	No	Description	Status
7	7.3.1	31	10	Implementations **MUST** be able to process multiple header field rows with the same name in any combination of the single-value-per-line or comma-separated value forms.	M
7	7.3.1	32	11	Even though an arbitrary number of parameter pairs may be attached to a header field value, any given parameter-name**MUST NOT** appear more than once.	M
7	7.3.2	32	12	If a header field appears in a message not matching its category (such as a request header field in a response), it**MUST** be ignored. Section 20 defines the classification of each header field.	M
7	7.3.3	33	13	Implementations **MUST** accept both the long and short forms of each header name.	M
7	7.4.1	33	14	The Internet media type of the message body **MUST** be given by the Content-Type header field.	M
7	7.4.1	33	15	If the body has undergone any encoding such as compression, then this**MUST** be indicated by the Content- Encoding header field; otherwise, Content-Encoding **MUST** be omitted.	M
			16		M
7	7.4.1	33	17	Implementations that send requests containing multipart message bodies **MUST** send a session description as a non-multipart message body if the remote implementation requests this through an Accept header field that does not contain multipart.	M
7	7.4.2	33	18	The "chunked" transfer encoding of HTTP/1.1**MUST NOT** be used for SIP.	M

Section	SubSection	Page	No	Description	Status
7	7.5	34	19	Implementations processing SIP messages over stream-oriented transports **MUST** ignore any CRLF appearing before the start-line [H4.1].	M

Section	SubSection	Page	No	Description	Status
8	8.1.1	35	1	A valid SIP request formulated by a UAC**MUST**, at a minimum, contain the following header fields: To, From, CSeq, Call-ID, Max-Forwards, and Via; all of these header fields are mandatory in all SIP requests.	M
8	8.1.1.1	35	2	The initial Request-URI of the message**SHOULD** be set to the value of the URI in the To field.	S
8	8.1.1.1	35	3	When a provider wishes to configure a UA with an outbound proxy, it is **RECOMMENDED** that this be done by providing it with a pre-existing route set with a single URI, that of the outbound proxy.	R
8	8.1.1.1	35	4	When a pre-existing route set is present, the procedures for populating the Request-URI and Route header field detailed in Section 12.2.1.1**MUST** be followed (even though there is no dialog), using the desired Request-URI as the remote target URI.	M
8	8.1.1.2	36	5	All SIP implementations**MUST** support the SIP URI scheme.	M
8	8.1.1.2	36	6	Any implementation that supports TLS**MUST** support the SIPS URI scheme.	M
8	8.1.1.2	36	7	A request outside of a dialog**MUST NOT** contain a To tag; the tag in the To field of a request identifies the peer of the dialog.	M
8	8.1.1.3	37	8	The From header field allows for a display name. A UAC**SHOULD** use the display name "Anonymous", along with a syntactically correct, but otherwise meaningless URI (like sip:thisis@anonymous.invalid), if the identity of the client is to remain hidden.	S
8	8.1.1.3	37	9	The From field **MUST** contain a new "tag" parameter, chosen by the UAC.	M

Section	SubSection	Page	No	Description	Status
8	8.1.1.4	37	10	It **MUST** be the same for all requests and responses sent by either UA in a dialog.	M
8	8.1.1.4	37	11	It **SHOULD** be the same in each registration from a UA.	S
8	8.1.1.4	37	12	In a new request created by a UAC outside of any dialog, the Call-ID header field **MUST** be selected by the UAC as a globally unique identifier over space and time unless overridden by method-specific behavior.	M
8	8.1.1.4	38	13	Use of cryptographically random identifiers (RFC 1750 [12]) in the generation of Call-IDs is **RECOMMENDED**. Implementations MAY use the form "localid@host".	R
8	8.1.1.5	38	14	The method **MUST** match that of the request. For non-REGISTER requests outside of a dialog, the sequence number value is arbitrary.	M
8	8.1.1.5	38	15	The sequence number value **MUST** be expressible as a 32-bit unsigned integer and **MUST** be less than 2**31.	M
8			16		M
8	8.1.1.6	39	17	A UAC **MUST** insert a Max-Forwards header field into each request it originates with a value that **SHOULD** be 70.	M
8			18		S

Section	SubSection	Page	No	Description	Status
8	8.1.1.7	39	19	When the UAC creates a request, it **MUST** insert a Via into that request.	M
8	8.1.1.7	39	20	The protocol name and protocol version in the header field **MUST** be SIP and 2.0, respectively.	M
8	8.1.1.7	39	21	The Via header field value **MUST** contain a branch parameter.	M
8	8.1.1.7	39	22	The branch parameter value **MUST** be unique across space and time for all requests sent by the UA.	M
8	8.1.1.7	39	23	The branch ID inserted by an element compliant with this specification **MUST** always begin with the characters "z9hG4bK".	M
8	8.1.1.8	40	24	The Contact header field **MUST** be present and contain exactly one SIP or SIPS URI in any request that can result in the establishment of a dialog.	M
8	8.1.1.8	40	25	That is, the Contact header field value contains the URI at which the UA would like to receive requests, and this URI **MUST** be valid even if used in subsequent requests outside of any dialogs.	M
8	8.1.1.8	40	26	If the Request-URI or top Route header field value contains a SIPS URI, the Contact header field **MUST** contain a SIPS URI as well.	M
8	8.1.1.9	40	27	If the UAC supports extensions to SIP that can be applied by the server to the response, the UAC **SHOULD** include a Supported header field in the request listing the option tags (Section 19.2) for those extensions.	S

Section	SubSection	Page	No	Description	Status
8	8.1.1.9	40	28	The option tags listed **MUST** only refer to extensions defined in standards-track RFCs.	M
8	8.1.1.9	40	29	If the UAC wishes to insist that a UAS understand an extension that the UAC will apply to the request in order to process the request, it **MUST** insert a Require header field into the request listing the option tag for that extension.	M
8	8.1.1.9	41	30	If the UAC wishes to apply an extension to the request and insist that any proxies that are traversed understand that extension, it **MUST** insert a Proxy-Require header field into the request listing the option tag for that extension.	M
8	8.1.1.9	41	31	As with the Supported header field, the option tags in the Require and Proxy-Require header fields **MUST** only refer to extensions defined in standards-track RFCs.	M
8	8.1.2	41	32	Unless there is local policy specifying otherwise, the destination **MUST** be determined by applying the DNS procedures described in [4] as follows.	M
8	8.1.2	41	33	If the first element in the route set indicated a strict router (resulting in forming the request as described in Section 12.2.1.1), the procedures **MUST** be applied to the Request-URI of the request.	M
8	8.1.2	41	34	Independent of which URI is used as input to the procedures of [4], if the Request-URI specifies a SIPS resource, the UAC **MUST** follow the procedures of [4] as if the input URI were a SIPS URI.	M
8	8.1.2	41	35	Local policy MAY specify an alternate set of destinations to attempt. If the Request-URI contains a SIPS URI, any alternate destinations **MUST** be contacted with TLS.	M
8	8.1.2	41	36	This provides a simple alternative to a pre-existing route set as a way to specify an outbound proxy. However, that approach for configuring an outbound proxy is **NOT RECOMMENDED**; a pre-existing route set with a single URI **SHOULD** be used instead.	R

Section	SubSection	Page	No	Description	Status
8	8.1.2	41	37		S
8	8.1.2	41	38	If the request contains a Route header field, the request **SHOULD** be sent to the locations derived from its topmost value, but MAY be sent to any server that the UA is certain will honor the Route and Request-URI policies specified in this document (as opposed to those in RFC 2543).	S
8	8.1.2	42	39	In particular, a UAC configured with an outbound proxy **SHOULD** attempt to send the request to the location indicated in the first Route header field value instead of adopting the policy of sending all messages to the outbound proxy.	S
8	8.1.2	42	40	The UAC **SHOULD** follow the procedures defined in [4] for stateful elements, trying each address until a server is contacted. Each try constitutes a new transaction, and therefore each carries a different topmost Via header field value with a new branch parameter.	S
8	8.1.3.1	42	41	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.	M
8	8.1.3.1	42	42	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.	M
8	8.1.3.2	42	43	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.	M
8			44		M
8	8.1.3.2	42	45	A UAC **MUST** treat any provisional response different than 100 that it does not recognize as 183 (Session Progress).	M

Section	SubSection	Page	No	Description	Status
8	8.1.3.2	42	46	A UAC **MUST** be able to process 100 and 183 responses.	M
8	8.1.3.3	43	47	If more than one Via header field value is present in a response, the UAC **SHOULD** discard the message.	S
8	8.1.3.4	43	48	Upon receipt of a redirection response (for example, a 301 response status code), clients **SHOULD** use the URI(s) in the Contact header field to formulate one or more new requests based on the redirected request.	S
8	8.1.3.4	43	49	As with proxy recursion, a client processing 3xx class responses **MUST NOT** add any given URI to the target set more than once.	M
8	8.1.3.4	43	50	If the original request had a SIPS URI in the Request- URI, the client MAY choose to recurse to a non-SIPS URI, but **SHOULD** inform the user of the redirection to an insecure URI.	S
8	8.1.3.4	44	51	Failures **SHOULD** be detected through failure response codes (codes greater than 399); for network errors the client transaction will report any transport layer failures to the transaction user.	S
8	8.1.3.4	44	52	When a failure for a particular contact address is received, the client **SHOULD** try the next contact address.	S
8	8.1.3.4	44	53	In order to create a request based on a contact address in a 3xx response, a UAC **MUST** copy the entire URI from the target set into the Request-URI, except for the "method-param" and "header" URI parameters (see Section 19.1.1 for a definition of these parameters).	M
8	8.1.3.4	44	54	It is **RECOMMENDED** that the UAC reuse the same To, From, and Call-ID used in the original redirected request, but the UAC MAY also choose to update the Call-ID header field value for new requests, for example.	R

Section	SubSection	Page	No	Description	Status
8	8.1.3.4	44	55	Finally, once the new request has been constructed, it is sent using a new client transaction, and therefore **MUST** have a new branch ID in the top Via field as discussed in Section 8.1.1.7.	M
8	8.1.3.4	45	56	In all other respects, requests sent upon receipt of a redirect response **SHOULD** re-use the header fields and bodies of the original request.	S
8	8.1.3.5	45	57	If a 401 (Unauthorized) or 407 (Proxy Authentication Required) response is received, the UAC **SHOULD** follow the authorization procedures of Section 22.2 and Section 22.3 to retry the request with credentials.	S
8	8.1.3.5	45	58	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.	S
8	8.1.3.5	45	59	If a 415 (Unsupported Media Type) response is received (Section 21.4.13), the request contained media types not supported by the UAS. The UAC**SHOULD** retry sending the request, this time only using content with types listed in the Accept header field in the response, with encodings listed in the Accept-Encoding header field in the response, and with languages listed in the Accept-Language in the response.	S
8	8.1.3.5	45	60	If a 416 (Unsupported URI Scheme) response is received (Section 21.4.14), the Request-URI used a URI scheme not supported by the server. The client **SHOULD** retry the request, this time, using a SIP URI.	S
8	8.1.3.5	45	61	If a 420 (Bad Extension) response is received (Section 21.4.15), the request contained a Require or Proxy-Require header field listing an option-tag for a feature not supported by a proxy or UAS. The UAC**SHOULD** retry the request, this time omitting any extensions listed in the Unsupported header field in the response.	S
8	8.1.3.5	45	62	In all of the above cases, the request is retried by creating a new request with the appropriate modifications. This new request constitutes a new transaction and **SHOULD** have the same value of the Call-ID, To, and From of the previous request, but the CSeq should contain a new sequence number that is one higher than the previous.	S
8	8.2	46	63	If a request is accepted, all state changes associated with it**MUST** be performed.	M

Section	SubSection	Page	No	Description	Status
8	8.2	46	64	If it is rejected, all state changes **MUST NOT** be performed.	M
8	8.2	46	65	UASs **SHOULD** process the requests in the order of the steps that follow in this section (that is, starting with authentication, then inspecting the method, the header fields, and so on throughout the remainder of this section).	S
8	8.2.1	46	66	Once a request is authenticated (or authentication is skipped), the UAS**MUST** inspect the method of the request.	M
8	8.2.1	46	67	If the UAS recognizes but does not support the method of a request, it**MUST** generate a 405 (Method Not Allowed) response.	M
8	8.2.1	46	68	Procedures for generating responses are described in Section 8.2.6. The UAS **MUST** also add an Allow header field to the 405 (Method Not Allowed) response.	M
8	8.2.1	46	69	The Allow header field**MUST** list the set of methods supported by the UAS generating the message.	M
8	8.2.2	46	70	If a UAS does not understand a header field in a request (that is, the header field is not defined in this specification or in any supported extension), the server **MUST** ignore that header field and continue processing the message.	M
8	8.2.2	46	71	A UAS **SHOULD** ignore any malformed header fields that are not necessary for processing requests.	S
8	8.2.2.1	47	72	A UAS MAY apply any policy it wishes to determine whether to accept requests when the To header field is not the identity of the UAS. However, it is **RECOMMENDED** that a UAS accept requests even if they do not recognize the URI scheme (for example, a tel: URI) in the To header field, or if the To header field does not address a known or current user of this UAS.	R

Section	SubSection	Page	No	Description	Status
8	8.2.2.1	47	73	If, on the other hand, the UAS decides to reject the request, it**SHOULD** generate a response with a 403 (Forbidden) status code and pass it to the server transaction for transmission.	S
8	8.2.2.1	47	74	If the Request-URI uses a scheme not supported by the UAS, it**SHOULD** reject the request with a 416 (Unsupported URI Scheme) response.	S
8	8.2.2.1	47	75	If the Request-URI does not identify an address that the UAS is willing to accept requests for, it **SHOULD** reject the request with a 404 (Not Found) response.	S
8	8.2.2.2	47	76	If the request has no tag in the To header field, the UAS core**MUST** check the request against ongoing transactions.	M
8	8.2.2.2	47	77	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.	S
8	8.2.2.3	47	78	If a UAS does not understand an option-tag listed in a Require header field, it **MUST** respond by generating a response with status code 420 (Bad Extension).	M
8	8.2.2.3	47	79	status code 420 (Bad Extension). The UAS**MUST** add an Unsupported header field, and list in it those options it does not understand amongst those in the Require header field of the request.	M
8	8.2.2.3	48	80	Note that Require and Proxy-Require**MUST NOT** be used in a SIP CANCEL request, or in an ACK request sent for a non-2xx response.	M
8	8.2.2.3	48	81	These header fields **MUST** be ignored if they are present in these requests.	M

Section	SubSection	Page	No	Description	Status
8	8.2.2.3	48	82	An ACK request for a 2xx responseMUST contain only those Require and Proxy-Require values that were present in the initial request.	M
8	8.2.3	48	83	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 UASMUST reject the request with a 415 (Unsupported Media Type) response.	M
8	8.2.3	48	84	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.	M
8	8.2.3	48	85	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.	M
8	8.2.3	48	86	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.	M
8	8.2.4	48	87	A UAS that wishes to apply some extension when generating the response MUST NOT do so unless support for that extension is indicated in the Supported header field in the request.	M
8	8.2.4	48	88	If the desired extension is not supported, the serverSHOULD rely only on baseline SIP and any other extensions supported by the client.	S
8	8.2.4	48	89	In rare circumstances, where the server cannot process the request without the extension, the server MAY send a 421 (Extension Required) response. This response indicates that the proper response cannot be generated without support of a specific extension. The needed extension(s)MUST be included in a Require header field in the response.	M
8	8.2.4	48	90	This behavior isNOT RECOMMENDED, as it will generally break interoperability.	R

Section	SubSection	Page	No	Description	Status
8	8.2.4	48	91	Any extensions applied to a non-421 response**MUST** be listed in a Require header field included in the response.	M
8	8.2.4	48	92	Of course, the server **MUST NOT** apply extensions not listed in the Supported header field in the request.	M
8	8.2.6.1	49	93	One largely non-method-specific guideline for the generation of responses is that UASs **SHOULD NOT** issue a provisional response for a non-INVITE request.	S
8	8.2.6.1	49	94	Rather, UASs **SHOULD** generate a final response to a non-INVITE request as soon as possible. When a 100 (Trying) response is generated, any Timestamp header field	S
8	8.2.6.1	50	95	possible. When a 100 (Trying) response is generated, any Timestamp header field present in the request**MUST** be copied into this 100 (Trying) response.	M
8	8.2.6.1	50	96	If there is a delay in generating the response, the UAS**SHOULD** add a delay value into the Timestamp value in the response.	S
8	8.2.6.1	50	97	This value **MUST** contain the difference between the time of sending of the response and receipt of the request, measured in seconds.	M
8	8.2.6.2	50	98	The From field of the response **MUST** equal the From header field of the request.	M
8	8.2.6.2	50	99	The Call-ID header field of the response**MUST** equal the Call-ID header field of the request.	M

Section	SubSection	Page	No	Description	Status
8	8.2.6.2	50	100	The CSeq header field of the response **MUST** equal the CSeq field of the request.	M
8	8.2.6.2	50	101	The Via header field values in the response **MUST** equal the Via header field values in the request and **MUST** maintain the same ordering.	M
8			102		M
8	8.2.6.2	50	103	If a request contained a To tag in the request, the To header field in the response **MUST** equal that of the request.	M
8	8.2.6.2	50	104	However, if the To header field in the request did not contain a tag, the URI in the To header field in the response **MUST** equal the URI in the To header field; additionally, the UAS **MUST** add a tag to the To header field in the response (with the exception of the 100 (Trying) response, in which a tag MAY be present).	M
8			105		M
8	8.2.6.2	50	106	The same tag **MUST** be used for all responses to that request, both final and provisional (again excepting the 100 (Trying)).	M
8	8.2.7	51	107	o A stateless UAS **MUST NOT** send provisional (1xx) responses.	M
8	8.2.7	51	108	o A stateless UAS **MUST NOT** retransmit responses.	M

Section	SubSection	Page	No	Description	Status
8	8.2.7	51	109	o A stateless UAS **MUST** ignore ACK requests.	M
8	8.2.7	51	110	o A stateless UAS **MUST** ignore CANCEL requests.	M
8	8.2.7	51	111	o To header tags **MUST** be generated for responses in a stateless manner - in a manner that will generate the same tag for the same request consistently. For information on tag construction see Section 19.3.	M
8	8.3	52	112	For well-formed CANCEL requests, it **SHOULD** return a 2xx response.	S
8	8.3	52	113	However, redirect servers **MUST NOT** redirect a request to a URI equal to the one in the Request-URI; instead, provided that the URI does not point to itself, the server MAY proxy the request to the destination URI, or MAY reject it with a 404.	M
8	8.3	52	114	The "expires" parameter of a Contact header field value indicates how long the URI is valid. The value of the parameter is a number indicating seconds. If this parameter is not provided, the value of the Expires header field determines how long the URI is valid. Malformed values**SHOULD** be treated as equivalent to 3600.	S
8	8.3	53	115	Redirect servers **MUST** ignore features that are not understood (including unrecognized header fields, any unknown option tags in Require, or even method names) and proceed with the redirection of the request in question.	M

Section	SubSection	Page	No	Description	Status
9	9.1	53	1	A CANCEL request **SHOULD NOT** be sent to cancel a request other than INVITE.	S
9	9.1	54	2	The Request-URI, Call-ID, To, the numeric part of CSeq, and From header fields in the CANCEL request **MUST** be identical to those in the request being cancelled, including tags.	M
9	9.1	54	3	A CANCEL constructed by a client **MUST** have only a single Via header field value matching the top Via value in the request being cancelled.	M
9	9.1	54	4	However, the method part of the CSeq header field **MUST** have a value of CANCEL.	M
9	9.1	54	5	If the request being cancelled contains a Route header field, the CANCEL request **MUST** include that Route header field's values.	M
9	9.1	54	6	The CANCEL request **MUST NOT** contain any Require or Proxy-Require header fields.	M
9	9.1	54	7	Once the CANCEL is constructed, the client **SHOULD** check whether it has received any response (provisional or final) for the request being cancelled (herein referred to as the "original request").	S
9	9.1	54	8	If no provisional response has been received, the CANCEL request **MUST NOT** be sent; rather, the client **MUST** wait for the arrival of a provisional response before sending the request. If the original request has generated a final response, the	M
9	9.1	54	9		M

Section	SubSection	Page	No	Description	Status
9	9.1	54	10	sending the request. If the original request has generated a final response, the CANCEL **SHOULD NOT** be sent, as it is an effective no-op, since CANCEL has no effect on requests that have already generated a final response. When the client	S
9	9.1	54	11	The destination address, port, and transport for the CANCEL**MUST** be identical to those used to send the original request.	M
9	9.1	55	12	If there is no final response for the original request in 64*T1 seconds (T1 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.	S
9			13		S
9	9.2	55	14	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).	S
9	9.2	55	15	If the original request was an INVITE, the UAS**SHOULD** immediately respond to the INVITE with a 487 (Request Terminated).	S
9	9.2	55	16	This response is constructed following the procedures described in Section 8.2.6 noting that the To tag of the response to the CANCEL and the To tag in the response to the original request**SHOULD** be the same. The response to CANCEL is passed to the server transaction for transmission.	S

Section	SubSection	Page	No	Description	Status
10	10.1	57	1	The only requirement is that a registrar for some domain**MUST** be able to read and write data to the location service, and a proxy or a redirect server for that domain **MUST** be capable of reading that same data.	M
			2		M
10	10.2	57	3	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.	M
			4		M
10	10.2	57	5	The following header fields, except Contact,**MUST** be included in a REGISTER request. (Request-URI: To: From: Call-ID: CSeq:)	M
10		57	6	Request-URI: The "userinfo" and "@" components of the SIP URI**MUST NOT** be present.	M
10	10.2	57	7	To: This address-of-record **MUST** be a SIP URI or SIPS URI. From: The From header field contains the address-of-record of the person responsible for the registration. The value is the same as the To header field unless the request is a third- party registration.	M
10		58	8	Call-ID: All registrations from a UAC**SHOULD** use the same Call-ID header field value for registrations sent to a particular registrar.	S
10	10.2	58	9	CSeq: A UA **MUST** increment the CSeq value by one for each REGISTER request with the same Call-ID.	M

Section	SubSection	Page	No	Description	Status
10		58	10	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.	M
10	10.2	59	11	action: UACs **SHOULD NOT** use the "action" parameter.	S
10		59	12	expires: Malformed values **SHOULD** be treated as equivalent to 3600.	S
10	10.2.1	60	13	If the address-of-record in the To header field of a REGISTER request is a SIPS URI, then any Contact header field values in the request **SHOULD** also be SIPS URIs.	S
10	10.2.2	61	14	A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a REGISTER request. UAs **SHOULD** support this mechanism so that bindings can be removed before their expiration interval has passed.	S
10	10.2.2	61	15	The REGISTER-specific Contact header field value of "*" applies to all registrations, but it **MUST NOT** be used unless the Expires header field is present with a value of "0".	M
10	10.2.4	61	16	A UA **SHOULD NOT** refresh bindings set up by other UAs. The 200 (OK) response from the registrar contains a list of Contact fields enumerating all current bindings.	S
10	10.2.4	62	17	A UA **SHOULD** use the same Call-ID for all registrations during a single boot cycle.	S
10	10.2.4	62	18	Registration refreshes **SHOULD** be sent to the same network address as the original registration, unless redirected.	S

Section	SubSection	Page	No	Description	Status
10	10.2.6	62	19	If there is no configured registrar address, the UA**SHOULD** use the host part of the address- of-record as the Request-URI and address the request there, using the normal SIP server location mechanisms [4].	M
10	10.2.7	63	20	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.	S
10	10.3	63	21	A registrar **MUST** not generate 6xx responses.	M
10	10.3	63	22	Registrars **MUST** ignore the Record-Route header field if it is included in a REGISTER request.	M
10	10.3	63	23	Registrars **MUST NOT** include a Record-Route header field in any response to a REGISTER request.	M
10	10.3	63	24	REGISTER requests **MUST** be processed by a registrar in the order that they are received.	M
10	10.3	63	25	REGISTER requests **MUST** also be processed atomically, meaning that a particular REGISTER request is either processed completely or not at all.	M
10	10.3	63	26	Each REGISTER message **MUST** be processed independently of any other registration or binding changes.	M
10	10.3	64	27	1. The registrar inspects the Request-URI to determine whether it has access to bindings for the domain identified in the Request-URI. If not, and if the server also acts as a proxy server, the server **SHOULD** forward the request to the addressed domain, following the general behavior for proxying messages described in Section 16.	S

Section	SubSection	Page	No	Description	Status
10	10.3	64	28	2. To guarantee that the registrar supports any necessary extensions, the registrar **MUST** process the Require header field values as described for UASs in Section 8.2.2.	M
10	10.3	64	29	3. A registrar **SHOULD** authenticate the UAC.	S
10	10.3	64	30	4. The registrar **SHOULD** determine if the authenticated user is authorized to modify registrations for this address-of-record. For example, a registrar might consult an authorization database that maps user names to a list of addresses-of-record for which that user has authorization to modify bindings. If the authenticated user is not authorized to modify bindings, the registra**MUST** return a 403 (Forbidden) and skip the remaining steps.	S
			31		M
10	10.3	64	32	5. The registrar extracts the address-of-record from the To header field of the request. If the address-of-record is not valid for the domain in the Request-URI, the registrar **MUST** send a 404 (Not Found) response and skip the remaining steps. The URI **MUST** then be converted to a canonical form. To do that, all URI parameters **MUST** be removed (including the user-param), and any escaped characters **MUST** be converted to their unescaped form. The result serves as an index into the list of bindings.	M
			33		M
			34		M
			35		M
10	10.3	65	36	6. The registrar checks whether the request contains the Contact header field. If not, it skips to the last step. If the Contact header field is present, the registrar checks if there is one Contact field value that contains the special value "*" and an Expires field. If the request has additional Contact fields or an expiration time other than zero, the request is invalid, and the serve**MUST** return a 400 (Invalid Request) and skip the remaining steps. If not, the registrar checks whether the Call-ID agrees with the value stored for each binding. If not, i**MUST** remove the	M

Section	SubSection	Page	No	Description	Status
			37	Call-ID agrees with the value stored for each binding. If not, it **MUST** remove the binding. If it does agree, it **MUST** remove the binding only if the CSeq in the request is higher than the value stored for that binding. Otherwise, the update **MUST** be aborted and the request fails.	M
			38		M
			39		M
10	10.3	65	40	7. The registrar now processes each contact address in the Contact header field in turn. For each address, it determines the expiration interval as follows: - If the field value has an "expires" parameter, that value **MUST** be taken as the requested expiration. - If there is no such parameter, but the request has an Expires header field, that value **MUST** be taken as the requested expiration. - If there is neither, a locally-configured default value **MUST** be taken as the requested expiration.	M
			41		M
			42		M
10	10.3	65	43	If and only if the requested expiration interval is greater than zero AND smaller than one hour AND less than a registrar-configured minimum, the registrar MAY reject the registration with a response of 423 (Interval Too Brief). This response **MUST** contain a Min-Expires header field that states the minimum expiration interval the registrar is willing to honor. It then skips the remaining steps.	M
10	10.3	66	44	If the binding does exist, the registrar checks the Call-ID value. If the Call-ID value in the existing binding differs from the Call-ID value in the request, the binding **MUST** be removed if the expiration time is zero and updated otherwise.	M
10	10.3	66	45	If they are the same, the registrar compares the CSeq value. If the value is higher than that of the existing binding, i**MUST** update or remove the binding as above. If not, the update **MUST** be aborted and the request fails.	M

Section	SubSection	Page	No	Description	Status
			46		M
10	10.3	66	47	The binding updates **MUST** be committed (that is, made visible to the proxy or redirect server) if and only if all binding updates and additions succeed.	M
10	10.3	66	48	If any one of them fails (for example, because the back-end database commit failed), the request **MUST** fail with a 500 (Server Error) response and all tentative binding updates **MUST** be removed.	M
			49		M
10	10.3	66	50	8. The registrar returns a 200 (OK) response. The response **MUST** contain Contact header field values enumerating all current bindings.	M
10	10.3	66	51	Each Contact value **MUST** feature an "expires" parameter indicating its expiration interval chosen by the registrar.	M
10	10.3	66	52	The response **SHOULD** include a Date header field.	S

Section	SubSection	Page	No	Description	Status
11		66	1	All UAs **MUST** support the OPTIONS method.	M
11	11.1	67	2	An Accept header field **SHOULD** be included to indicate the type of message body the UAC wishes to receive in the response.	S
11	11.2	68	3	The response code chosen **MUST** be the same that would have been chosen had the request been an INVITE.	M
11	11.2	68	4	Allow, Accept, Accept-Encoding, Accept-Language, and Supported header fields **SHOULD** be present in a 200 (OK) response to an OPTIONS request.	S
11	11.2	68	5	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.	S
11	11.2	68	6	If the types include one that can describe media capabilities, the UAS **SHOULD** include a body in the response for that purpose.	S

Section	SubSection	Page	No	Description	Status
12	12.1	70	1	UAs **MUST** assign values to the dialog ID components as described below.	M
12	12.1.1	70	2	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)	M
12	12.1.1	70	3	(When a UAS responds to a request with a response that establishes a dialog (such as a 2xx to INVITE)),the UAS**MUST** maintain the order of those values(including the URIs, URI parameters, and any Record-Route header field parameters, whether they are known or unknown to the UAS).	M
12	12.1.1	70	4	The UAS **MUST** add a Contact header field to the response.	M
12	12.1.1	70	5	The URI provided in the Contact header field**MUST** be a SIP or SIPS URI.	M
12	12.1.1	71	6	If the request that initiated the dialog contained a SIPS URI in the Request-URI or in the top Record-Route header field value, if there was any, or the Contact header field if there was no Record-Route header field, the Contact header field in the response **MUST** be a SIPS URI.	M
12	12.1.1	71	7	The URI **SHOULD** have global scope (that is, the same URI can be used in messages outside this dialog).	S
12	12.1.1	71	8	The UAS then constructs the state of the dialog. This state**MUST** be maintained for the duration of the dialog.	M
12	12.1.1	71	9	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.	M

Section	SubSection	Page	No	Description	Status
12	12.1.1	71	10	If no Record-Route header field is present in the request, the route set**MUST** be set to the empty set.	M
12	12.1.1	71	11	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.	M
12	12.1.1	71	12	The remote sequence number **MUST** be set to the value of the sequence number in the CSeq header field of the request.	M
12	12.1.1	71	13	The local sequence number**MUST** be empty.	M
12	12.1.1	71	14	The call identifier component of the dialog ID**MUST** be set to the value of the Call-ID in the request.	M
12	12.1.1	71	15	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),	M
12	12.1.1	71	16	the remote tag component of the dialog ID **MUST** be set to the tag from the From field in the request.	M
12	12.1.1	71	17	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.	M
12	12.1.1	71	18	The remote URI **MUST** be set to the URI in the From field,	M

Section	SubSection	Page	No	Description	Status
12	12.1.1	71	19	the local URI MUST be set to the URI in the To field.	M
12	12.1.2	71	20	When a UAC sends a request that can establish a dialog (such as an INVITE) it MUST provide a SIP or SIPS URI with global scope (i.e., the same SIP URI can be used in messages outside this dialog) in the Contact header field of the request.	M
12	12.1.2	71	21	If the request has a Request- URI or a topmost Route header field value with a SIPS URI, the Contact header field MUST contain a SIPS URI	M
12	12.1.2	72	22	When a UAC receives a response that establishes a dialog, it constructs the state of the dialog. This state MUST be maintained for the duration of the dialog.	M
12	12.1.2	72	23	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.	M
12	12.1.2	72	24	If no Record-Route header field is present in the response, the route set MUST be set to the empty set.	M
12	12.1.2	72	25	The remote target MUST be set to the URI from the Contact header field of the response.	M
12	12.1.2	72	26	The local sequence number MUST be set to the value of the sequence number in the CSeq header field of the request.	M
12	12.1.2	72	27	The remote sequence number MUST be empty (it is established when the remote UA sends a request within the dialog).	M

Section	SubSection	Page	No	Description	Status
12	12.1.2	72	28	The call identifier component of the dialog ID**MUST** be set to the value of the Call-ID in the request.	M
12	12.1.2	72	29	The local tag component of the dialog ID**MUST** be set to the tag in the From field in the request,	M
12	12.1.2	72	30	the remote tag component of the dialog ID **MUST** be set to the tag in the To field of the response.	M
12	12.1.2	72	31	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.	M
12	12.1.2	72	32	The remote URI **MUST** be set to the URI in the To field,	M
12	12.1.2	72	33	the local URI **MUST** be set to the URI in the From field.	M
12	12.2.1.1	73	34	The URI in the To field of the request**MUST** be set to the remote URI from the dialog state.	M
12	12.2.1.1	73	35	The tag in the To header field of the request**MUST** be set to the remote tag of the dialog ID.	M
12	12.2.1.1	73	36	The From URI of the request **MUST** be set to the local URI from the dialog state.	M

Section	SubSection	Page	No	Description	Status
12	12.2.1.1	73	37	The tag in the From header field of the request**MUST** be set to the local tag of the dialog ID.	M
12	12.2.1.1	73	38	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.	M
12	12.2.1.1	73	39	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.	M
12	12.2.1.1	73	40	The Call-ID of the request **MUST** be set to the Call-ID of the dialog.	M
12	12.2.1.1	73	41	Requests within a dialog **MUST** contain strictly monotonically increasing and contiguous CSeq sequence numbers (increasing-by-one) in each direction (excepting ACK and CANCEL of course, whose numbers equal the requests being acknowledged or cancelled).	M
12	12.2.1.1	73	42	Therefore, if the local sequence number is not empty, the value of the local sequence number **MUST** be incremented by one,	M
12	12.2.1.1	73	43	this value **MUST** be placed into the CSeq header field.	M
12	12.2.1.1	73	44	If the local sequence number is empty, an initial value**MUST** be chosen using the guidelines of Section 8.1.1.5.	M
12	12.2.1.1	73	45	The method field in the CSeq header field value**MUST** match the method of the request.	M

Section	SubSection	Page	No	Description	Status
12	12.2.1.1	74	46	If the route set is empty, the UAC **MUST** place the remote target URI into the Request-URI.	M
12	12.2.1.1	74	47	The UAC **MUST NOT** add a Route header field to the request.	M
12	12.2.1.1	74	48	If the route set is not empty, and the first URI in the route set contains the lr parameter (see Section 19.1.1), the UAC **MUST** place the remote target URI into the Request-URI	M
12	12.2.1.1	74	49	(If the route set is not empty, and the first URI in the route set contains the lr parameter (see Section 19.1.1),) the UAC **MUST** include a Route header field containing the route set values in order, including all parameters.	M
12	12.2.1.1	74	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.	M
12	12.2.1.1	74	51	The UAC **MUST** add a Route header field containing the remainder of the route set values in order, including all parameters.	M
12	12.2.1.1	74	52	The UAC **MUST** then place the remote target URI into the Route header field as the last value.	M
12	12.2.1.1	75	53	A UAC **SHOULD** include a Contact header field in any target refresh requests within a dialog, and unless there is a need to change it,	S
12	12.2.1.1	75	54	the URI **SHOULD** be the same as used in previous requests within the dialog.	S

Section	SubSection	Page	No	Description	Status
12	12.2.1.1	75	55	If the "secure" flag is true, that URI **MUST** be a SIPS URI.	M
12	12.2.1.2	75	56	When a UAC receives a 2xx response to a target refresh request, it**MUST** replace the dialog's remote target URI with the URI from the Contact header field in that response, if present.	M
12	12.2.1.2	76	57	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.	S
12	12.2.1.2	76	58	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.)	S
12	12.2.2	76	59	If the UAS wishes to reject the request because it does not wish to recreate the dialog, it **MUST** respond to the request with a 481 (Call/Transaction Does Not Exist) status code and pass that to the server transaction.	M
12	12.2.2	77	60	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.	M
12	12.2.2	77	61	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.	M
12	12.2.2	77	62	It is possible for the CSeq sequence number to be higher than the remote sequence number by more than one. This is not an error condition, and a UAS **SHOULD** be prepared to receive and process requests with CSeq values more than one higher than the previous received request.	S
12	12.2.2	77	63	The UAS **MUST** then set the remote sequence number to the value of the sequence number in the CSeq header field value in the request.	M

Section	SubSection	Page	No	Description	Status
12	12.2.2	77	64	When a UAS receives a target refresh request, it **MUST** replace the dialog's remote target URI with the URI from the Contact header field in that request, if present.	M

Section	SubSection	Page	No	Description	Status
13	13.1	78	1	A UA that supports INVITE **MUST** also support ACK, CANCEL and BYE.	M
	13.2.1	78	2	An Allow header field (Section 20.5) **SHOULD** be present in the INVITE.	S
	13.2.1	78	3	For example, a UA capable of receiving INFO requests within a dialog [34] **SHOULD** include an Allow header field listing the INFO method.	S
	13.2.1	78	4	A Supported header field (Section 20.37) **SHOULD** be present in the INVITE. It enumerates all the extensions understood by the UAC.	S
	13.2.1	79	5	If the time indicated in the Expires header field is reached and no final answer for the INVITE has been received, the UAC core **SHOULD** generate a CANCEL request for the INVITE, as per Section 9.	S
	13.2.1	79	6	o The initial offer **MUST** be in either an INVITE or, if not there, in the first reliable non-failure message from the UAS back to the UAC. In this specification, that is the final 2xx response.	M
	13.2.1	80	7	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.	M
	13.2.1	80	8	The UAC **MUST** treat the first session description it receives as the answer,	M
	13.2.1	80	9	(The UAC) **MUST** ignore any session descriptions in subsequent responses to the initial INVITE.	M

Section	SubSection	Page	No	Description	Status
	13.2.1	80	10	o If the initial offer is in the first reliable non-failure message from the UAS back to UAC, the answer **MUST** be in the acknowledgement for that message (in this specification, ACK for a 2xx response).	M
	13.2.1	80	11	o Once the UAS has sent or received an answer to the initial offer, i**MUST NOT** generate subsequent offers in any responses to the initial INVITE. This means that a UAS based on this specification alone can never generate subsequent offers until completion of the initial transaction.	M
	13.2.1	80	12	Concretely, the above rules specify two exchanges for UAs compliant to this specification alone - the offer is in the INVITE, and the answer in the 2xx (and possibly in a 1xx as well, with the same value), or the offer is in the 2xx, and the answer is in the ACK. All user agents that support INVITE**MUST** support these two exchanges.	M
	13.2.1	80	13	The Session Description Protocol (SDP) (RFC 2327 [1])**MUST** be supported by all user agents as a means to describe sessions	M
	13.2.1	80	14	and its usage for constructing offers and answers**MUST** follow the procedures defined in [13].	M
	13.2.2.3	81	15	Subsequent final responses (which would only arrive under error conditions) **MUST** be ignored.	M
	13.2.2.4	82	16	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,	M
	13.2.2.4	82	17	and the route set for the dialog **MUST** be recomputed based on the 2xx response using the procedures of Section 12.2.1.2.	M
	13.2.2.4	82	18	Otherwise, a new dialog in the "confirmed" state**MUST** be constructed using the procedures of Section 12.1.2.	M

Section	SubSection	Page	No	Description	Status
	13.2.2.4	82	19	The UAC core **MUST** generate an ACK request for each 2xx received from the transaction layer.	M
	13.2.2.4	82	20	The header fields of the ACK are constructed in the same way as for any request sent within a dialog (see Section 12) with the exception of the CSeq and the header fields related to authentication.	M
	13.2.2.4	82	21	The sequence number of the CSeq header field**MUST** be the same as the INVITE being acknowledged,	M
	13.2.2.4	82	22	(The sequence number of the CSeq header field MUST be the same as the INVITE being acknowledged,)but the CSeq method MUST be ACK. The ACK **MUST** contain the same credentials as the INVITE.	M
	13.2.2.4	82	23	If the 2xx contains an offer (based on the rules above), the ACK**MUST** carry an answer in its body.	M
	13.2.2.4	82	24	If the offer in the 2xx response is not acceptable, the UAC core**MUST** generate a valid answer in the ACK and then send a BYE immediately.	M
	13.2.2.4	82	25	The ACK **MUST** be passed to the client transport every time a retransmission of the 2xx final response that triggered the ACK arrives.	M
	13.2.2.4	83	26	If, after acknowledging any 2xx response to an INVITE, the UAC does not want to continue with that dialog, then the UAC**MUST** terminate the dialog by sending a BYE request as described in Section 15.	M
	13.3.1	83	27	1. If the request is an INVITE that contains an Expires header 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.	S

Section	SubSection	Page	No	Description	Status
	13.3.1	84	28	A UAS MAY send as many provisional responses as it likes. Each of these **MUST** indicate the same dialog ID.	M
	13.3.1.1	84	29	Each of these **MUST** indicate the same dialog ID. However, these will not be delivered reliably.	M
	13.3.1.1	84	30	To prevent cancellation, the UAS**MUST** send a non-100 provisional response at every minute, to handle the possibility of lost provisional responses.	M
	13.3.1.2	85	31	If the UAS decides to redirect the call, a 3xx response is sent. A 300 (Multiple Choices), 301 (Moved Permanently) or 302 (Moved Temporarily) response **SHOULD** contain a Contact header field containing one or more URIs of new addresses to be tried. The response is passed to the INVITE server transaction, which will deal with its retransmissions.	S
	13.3.1.3	85	32	A common scenario occurs when the callee is currently not willing or able to take additional calls at this end system. A 486 (Busy Here)**SHOULD** be returned in such a scenario.	S
	13.3.1.3	85	33	If the UAS knows that no other end system will be able to accept this call, a 600 (Busy Everywhere) response **SHOULD** be sent instead.	S
	13.3.1.3	85	34	A UAS rejecting an offer contained in an INVITE**SHOULD** return a 488 (Not Acceptable Here) response.	S
	13.3.1.3	85	35	Such a response **SHOULD** include a Warning header field value explaining why the offer was rejected.	S
	13.3.1.4	85	36	A 2xx response to an INVITE**SHOULD** contain the Allow header field and the Supported header field, and MAY contain the Accept header field.	S

Section	SubSection	Page	No	Description	Status
	13.3.1.4	85	37	If the INVITE request contained an offer, and the UAS had not yet sent an answer, the 2xx **MUST** contain an answer.	M
	13.3.1.4	85	38	If the INVITE did not contain an offer, the 2xx**MUST** contain an offer if the UAS had not yet sent an offer.	M
	13.3.1.4	86	39	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.	S

Section	SubSection	Page	No	Description	Status
14		86	1	The behavior of a UA on detection of media failure is a matter of local policy. However, automated generation of re-INVITE or BYE is**NOT RECOMMENDED** to avoid flooding the network with traffic when there is congestion.	R
14		86	2	In any case, if these messages are sent automatically, they**SHOULD** be sent after some randomized interval.(these messages = re-INVITE,BYE)	S
14	14.1	87	3	If the session description format has the capability for version numbers, the offerer **SHOULD** indicate that the version of the session description has changed.	S
14	14.1	87	4	Note that a UAC **MUST NOT** initiate a new INVITE transaction within a dialog while another INVITE transaction is in progress in either direction.	M
14	14.1	87	5	1. If there is an ongoing INVITE client transaction, the TU**MUST** wait until the transaction reaches the completed or terminated state before initiating the new INVITE.	M
14	14.1	87	6	2. If there is an ongoing INVITE server transaction, the TU**MUST** wait until the transaction reaches the confirmed or terminated state before initiating the new INVITE.	M
14	14.1	87	7	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.	M
14	14.1	88	8	If a UAC receives a 491 response to a re-INVITE, it**SHOULD** start a timer with a value T chosen as follows:	S
14	14.1	88	9	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.	S

Section	SubSection	Page	No	Description	Status
14	14.2	88	10	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	M
14	14.2	88	11	(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** include a Retry-After header field with a randomly chosen value of between 0 and 10 seconds.	M
14	14.2	88	12	A UAS that receives an INVITE on a dialog while an INVITE it had sent on that dialog is in progress **MUST** return a 491 (Request Pending) response to the received INVITE.	M
14	14.2	88	13	If a UA receives a re-INVITE for an existing dialog, it **MUST** check any version identifiers in the session description or, if there are no version identifiers, the content of the session description to see if it has changed.	M
14	14.2	88	14	If the session description has changed, the UAS **MUST** adjust the session parameters accordingly, possibly after asking the user for confirmation.	M
14	14.2	89	15	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.	S
14	14.2	89	16	If a UAS generates a 2xx response and never receives an ACK, it **SHOULD** generate a BYE to terminate the dialog.	S
14	14.2	89	17	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.	S
14	14.2	89	18	Specifically, this means that it **SHOULD** include as many media formats and media types that the UA is willing to support.	S

Section	SubSection	Page	No	Description	Status
14	14.2	89	19	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.	M
14	14.2	89	20	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.	S

Section	SubSection	Page	No	Description	Status
15		89	1	When a BYE is received on a dialog, any session associated with that dialog **SHOULD** terminate.	S
15		89	2	A UA **MUST NOT** send a BYE outside of a dialog.	M
15		89	3	the callee's UA MAY send a BYE on confirmed dialogs, but**MUST NOT** send a BYE on early dialogs.	M
15		90	4	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.	M
15	15.1.1	90	5	The UAC **MUST** consider the session terminated (and therefore stop sending or listening for media) as soon as the BYE request is passed to the client transaction.	M
15	15.1.1	91	6	If the response for the BYE is a 481 (Call/Transaction Does Not Exist) or a 408 (Request Timeout) or no response at all is received for the BYE (that is, a timeout is returned by the client transaction), the UAC**MUST** consider the session and the dialog terminated.	M
15	15.1.2	91	7	If the BYE does not match an existing dialog, the UAS core**SHOULD** generate a 481 (Call/Transaction Does Not Exist) response and pass that to the server transaction.	S
15	15.1.2	91	8	A UAS core receiving a BYE request for an existing dialog**MUST** follow the procedures of Section 12.2.2 to process the request.	M
15	15.1.2	91	9	Once done, the UAS **SHOULD** terminate the session (and therefore stop sending and listening for media).	S

Section	SubSection	Page	No	Description	Status
15	15.1.2	91	10	Whether or not it ends its participation on the session, the UAS core**MUST** generate a 2xx response to the BYE	M
15	15.1.2	91	11	(the UAS core) **MUST** pass that to the server transaction for transmission.	M
15	15.1.2	91	12	The UAS **MUST** still respond to any pending requests received for that dialog.	M
15	15.1.2	91	13	It is **RECOMMENDED** that a 487 (Request Terminated) response be generated to those pending requests.	R

Section	SubSection	Page	No	Description	Status
16	16.1	92	1	When responding directly to a request, the element is playing the role of a UAS and **MUST** behave as described in Section 8.2.	M
16	16.1	92	2	A stateful proxy MAY choose to "fork" a request, routing it to multiple destinations. Any request that is forwarded to more than one location**MUST** be handled statefully.	M
16	16.1	92	3	Requests forwarded between different types of transports where the proxy's TU must take an active role in ensuring reliable delivery on one of the transports **MUST** be forwarded transaction statefully.	M
16	16.1	92	4	A stateful proxy MAY transition to stateless operation at any time during the processing of a request, so long as it did not do anything that would otherwise prevent it from being stateless initially (forking, for example, or generation of a 100 response). When performing such a transition, all state is simply discarded. The proxy **SHOULD NOT** initiate a CANCEL request.	S
16	16.2	93	5	A stateful proxy creates a new server transaction for each new request received. Any retransmissions of the request will then be handled by that server transaction per Section 17. The proxy core**MUST** behave as a UAS with respect to sending an immediate provisional on that server transaction (such as 100 Trying) as described in Section 8.2.6. Thus, a stateful proxy**SHOULD NOT** generate 100 (Trying) responses to non-INVITE requests.	M
16			6		S
16	16.2	93	7	For all new requests, including any with unknown methods, an element intending to proxy the request**MUST**: 1. Validate the request (Section 16.3) 2. Preprocess routing information (Section 16.4) 3. Determine target(s) for the request (Section 16.5) 4. Forward the request to each target (Section 16.6) 5. Process all responses (Section 16.7)	M
16	16.3	94	8	Before an element can proxy a request, it**MUST** verify the message's validity. A valid message must pass the following checks: 1. Reasonable Syntax 2. URI scheme 3. Max-Forwards 4. (Optional) Loop Detection 5. Proxy-Require	M
16	16.3	94	9	If any of these checks fail, the element**MUST** behave as a user agent server (see Section 8.2) and respond with an error code.	M

Section	SubSection	Page	No	Description	Status
16	16.3	94	10	Notice that a proxy is not required to detect merged requests and**MUST NOT** treat merged requests as an error condition.	M
16	16.3	94	11	The request **MUST** be well-formed enough to be handled with a server transaction.	M
16	16.3	94	12	Any components involved in the remainder of these Request Validation steps or the Request Forwarding section **MUST** be well-formed.	M
16	16.3	94	13	Any other components, well-formed or not,**SHOULD** be ignored and remain unchanged when the message is forwarded.	S
16	16.3	94	14	This protocol is designed to be extended. Future extensions may define new methods and header fields at any time. An element**MUST NOT** refuse to proxy a request because it contains a method or header field it does not know about.	M
16	16.3	95	15	If the Request-URI has a URI whose scheme is not understood by the proxy, the proxy **SHOULD** reject the request with a 416 (Unsupported URI Scheme) response.	S
16	16.3	95	16	If the request contains a Max-Forwards header field with a field value of zero (0), the element **MUST NOT** forward the request.	M
16	16.3	95	17	If the request was for OPTIONS, the element MAY act as the final recipient and respond per Section 11. Otherwise, the element**MUST** return a 483 (Too many hops) response.	M
16	16.3	96	18	If the request contains a Proxy-Require header field (Section 20.29) with one or more option-tags this element does not understand, the element**MUST** return a 420 (Bad Extension) response.	M

Section	SubSection	Page	No	Description	Status
16	16.3	96	19	The response **MUST** include an Unsupported (Section 20.40) header field listing those option-tags the element did not understand.	M
16	16.3	96	20	If an element requires credentials before forwarding a request, the request **MUST** be inspected as described in Section 22.3. That section also defines what the element must do if the inspection fails.	M
16	16.4	96	21	The proxy **MUST** inspect the Request-URI of the request.	M
16	16.4	96	22	If the Request-URI of the request contains a value this proxy previously placed into a Record-Route header field (see Section 16.6 item 4), the proxy**MUST** replace the Request-URI in the request with the last value from the Route header field, and remove that value from the Route header field. The proxy **MUST** then proceed as if it received this modified request.	M
16	16.4	96	23		M
16	16.4	96	24	If the Request-URI contains a maddr parameter, the proxy**MUST** check to see if its value is in the set of addresses or domains the proxy is configured to be responsible for. If the Request-URI has a maddr parameter with a value the proxy is responsible for, and the request was received using the port and transport indicated (explicitly or by default) in the Request-URI, the proxy**MUST** strip the maddr and any non-default port or transport parameter and continue processing as if those values had not been present in the request.	M
16	16.4	96	25		M
16	16.4	97	26	If the first value in the Route header field indicates this proxy, the proxy**MUST** remove that value from the request.	M
16	16.5	97	27	If the Request-URI of the request contains an maddr parameter, the Request-URI **MUST** be placed into the target set as the only target URI, and the proxy **MUST** proceed to Section 16.6.	M

Section	SubSection	Page	No	Description	Status
16	16.5	97	28		M
16	16.5	97	29	If the domain of the Request-URI indicates a domain this element is not responsible for, the Request-URI **MUST** be placed into the target set as the only target, and the element **MUST** proceed to the task of Request Forwarding (Section 16.6).	M
16	16.5	97	30		M
16	16.5	97	31	When accessing the location service constructed by a registrar, the Request-URI **MUST** first be canonicalized as described in Section 10.3 before being used as an index.	M
16	16.5	97	32	If the Request-URI does not provide sufficient information for the proxy to determine the target set, it **SHOULD** return a 485 (Ambiguous) response. This response **SHOULD** contain a Contact header field containing URIs of new addresses to be tried.	S
16	16.5	97	33		S
16	16.5	98	34	As potential targets are located through these services, their URIs are added to the target set. Targets can only be placed in the target set once. If a target URI is already present in the set (based on the definition of equality for the URI type), it **MUST NOT** be added again.	M
16	16.5	98	35	A proxy **MUST NOT** add additional targets to the target set if the Request-URI of the original request does not indicate a resource this proxy is responsible for.	M
16	16.5	98	36	If a proxy uses a dynamic source of information while building the target set (for instance, if it consults a SIP Registrar), it**SHOULD** monitor that source for the duration of processing the request. New locations**SHOULD** be added to the target set as they become available. As above, any given URI**MUST NOT** be added to the set more than once.	S

Section	SubSection	Page	No	Description	Status
16			37		S
16			38		M
16	16.5	99	39	If the Request-URI indicates a resource at this proxy that does not exist, the proxy **MUST** return a 404 (Not Found) response.	M
16	16.5	99	40	If the target set remains empty after applying all of the above, the proxy**MUST** return an error response, which**SHOULD** be the 480 (Temporarily Unavailable) response.	M
16	16.5	99	41		S
16	16.6_1	100	42	The proxy starts with a copy of the received request. The copy**MUST** initially contain all of the header fields from the received request.	M
16	16.6_1	100	43	Fields not detailed in the processing described below**MUST NOT** be removed.	M
16	16.6_1	100	44	The copy **SHOULD** maintain the ordering of the header fields as in the received request.	S
16	16.6_1	100	45	The proxy **MUST NOT** reorder field values with a common field name (See Section 7.3.1).	M

Section	SubSection	Page	No	Description	Status
16	16.6_1	100	46	The proxy **MUST NOT** add to, modify, or remove the message body.	M
16	16.6_2	100	47	The Request-URI in the copy's start line**MUST** be replaced with the URI for this target.	M
16	16.6_2	100	48	If the URI contains any parameters not allowed in a Request-URI, they**MUST** be removed.	M
16	16.6_3	100	49	If the copy contains a Max-Forwards header field, the proxy**MUST** decrement its value by one (1).	M
16	16.6_3	100	50	If the copy does not contain a Max-Forwards header field, the proxy**MUST** add one with a field value, which**SHOULD** be 70.	M
16			51		S
16	16.6_4	101	52	If this proxy wishes to remain on the path of future requests in a dialog created by this request (assuming the request creates a dialog), it**MUST** insert a Record-Route header field value into the copy before any existing Record-Route header field values, even if a Route header field is already present.	M
16	16.6_4	101	53	If this request is already part of a dialog, the proxy**SHOULD** insert a Record-Route header field value if it wishes to remain on the path of future requests in the dialog. In normal endpoint operation as described in Section 12, these Record- Route header field values will not have any effect on the route sets used by the endpoints.	S
16	16.6_4	101	54	The URI placed in the Record-Route header field value**MUST** be a SIP or SIPS URI. This URI **MUST** contain an lr parameter (see Section 19.1.1).	M

Section	SubSection	Page	No	Description	Status
16			55		M
16	16.6_4	101	56	The URI **SHOULD NOT** contain the transport parameter unless the proxy has knowledge (such as in a private network) that the next downstream element that will be in the path of subsequent requests supports that transport.	S
16	16.6_4	102	57	The URI placed in the Record-Route header field**MUST** resolve to the element inserting it (or a suitable stand-in) when the server location procedures of [4] are applied to it, so that subsequent requests reach the same SIP element.	M
16	16.6_4	102	58	If the Request-URI contains a SIPS URI, or the topmost Route header field value (after the post processing of bullet 6) contains a SIPS URI, the URI placed into the Record-Route header field**MUST** be a SIPS URI. Furthermore, if the request was not received over TLS, the proxy**MUST** insert a Record-Route header field.	M
16			59		M
16	16.6_4	102	60	In a similar fashion, a proxy that receives a request over TLS, but generates a request without a SIPS URI in the Request-URI or topmost Route header field value (after the post processing of bullet 6),**MUST** insert a Record-Route header field that is not a SIPS URI.	M
16	16.6_4	102	61	If the URI placed in the Record-Route header field needs to be rewritten when it passes back through in a response, the URI**MUST** be distinct enough to locate at that time.	M
16	16.6_4	102	62	If a proxy needs to be in the path of any type of dialog (such as one straddling a firewall), it**SHOULD** add a Record-Route header field value to every request with a method it does not understand since that method may have dialog semantics.	S
16	16.6_4	102	63	Endpoints **MUST NOT** use a URI obtained from a Record-Route header field outside the dialog in which it was provided. See Section 12 for more information on an endpoint's use of Record-Route header fields.	M

Section	SubSection	Page	No	Description	Status
16	16.6_6	103	64	A proxy MAY have a local policy that mandates that a request visit a specific set of proxies before being delivered to the destination. A proxy**MUST** ensure that all such proxies are loose routers.	M
16	16.6_6	103	65	Generally, this can only be known with certainty if the proxies are within the same administrative domain. This set of proxies is represented by a set of URIs (each of which contains the lr parameter). This set**MUST** be pushed into the Route header field of the copy ahead of any existing values, if present. If the Route header field is absent, it**MUST** be added, containing that list of URIs.	M
16			66		M
16	16.6_6	103	67	If the proxy has a local policy that mandates that the request visit one specific proxy, an alternative to pushing a Route value into the Route header field is to bypass the forwarding logic of item 10 below, and instead just send the request to the address, port, and transport for that specific proxy. If the request has a Route header field, this alternative**MUST NOT** be used unless it is known that next hop proxy is a loose router.	M
16	16.6_6	103	68	Furthermore, if the Request-URI contains a SIPS URI, TLS**MUST** be used to communicate with that proxy.	M
16	16.6_6	103	69	If the copy contains a Route header field, the proxy**MUST** inspect the URI in its first value.	M
16	16.6_6	103	70	If that URI does not contain an lr parameter, the proxy**MUST** modify the copy as follows: - The proxy **MUST** place the Request-URI into the Route header field as the last value. - The proxy **MUST** then place the first Route header field value into the Request-URI and remove that value from the Route header field.	M
16	16.6_6	104	71		M
16	16.6_6	104	72		M

Section	SubSection	Page	No	Description	Status
16	16.6_7	104	73	The proxy MAY have a local policy to send the request to a specific IP address, port, and transport, independent of the values of the Route and Request-URI. Such a policy **MUST NOT** be used if the proxy is not certain that the IP address, port, and transport correspond to a server that is a loose router. However, this mechanism for sending the request through a specific next hop is **NOT RECOMMENDED**; instead a Route header field should be used for that purpose as described above.	M
16			74		R
16	16.6_7	104	75	If the proxy has reformatted the request to send to a strict-routing element as described in step 6 above, the proxy **MUST** apply those procedures to the Request-URI of the request.	M
16	16.6_7	104	76	Otherwise, the proxy **MUST** apply the procedures to the first value in the Route header field, if present, else the Request-URI.	M
16	16.6_7	104	77	Independently of which URI is being used as input to the procedures of [4], if the Request-URI specifies a SIPS resource, the proxy **MUST** follow the procedures of [4] as if the input URI were a SIPS URI.	M
16	16.6_7	104	78	As described in [4], the proxy **MUST** attempt to deliver the message to the first tuple in that set, and proceed through the set in order until the delivery attempt succeeds.	M
16	16.6_7	104	79	For each tuple attempted, the proxy **MUST** format the message as appropriate for the tuple and send the request using a new client transaction as detailed in steps 8 through 10.	M
16	16.6_7	105	80	Since each attempt uses a new client transaction, it represents a new branch. Thus, the branch parameter provided with the Via header field inserted in step 8 **MUST** be different for each attempt.	M
16	16.6_8	105	81	The proxy **MUST** insert a Via header field value into the copy before the existing Via header field values.	M

Section	SubSection	Page	No	Description	Status
16	16.6_8	105	82	Proxies choosing to detect loops have an additional constraint in the value they use for construction of the branch parameter. A proxy choosing to detect loops **SHOULD** create a branch parameter separable into two parts by the implementation. The first part **MUST** satisfy the constraints of Section 8.1.1.7 as described above. The second is used to perform loop detection and distinguish loops from spirals.	S
16			83		M
16	16.6_8	105	84	Loop detection is performed by verifying that, when a request returns to a proxy, those fields having an impact on the processing of the request have not changed. The value placed in this part of the branch parameter**SHOULD** reflect all of those fields (including any Route, Proxy-Require and Proxy-Authorization header fields).	S
16	16.6_8	106	85	If a proxy wishes to detect loops, the "branch" parameter it supplies**MUST** depend on all information affecting processing of a request, including the incoming Request-URI and any header fields affecting the request's admission or routing.	M
16	16.6_8	106	86	The request method **MUST NOT** be included in the calculation of the branch parameter.	M
16	16.6_8	106	87	In particular, CANCEL and ACK requests (for non-2xx responses)**MUST** have the same branch value as the corresponding request they cancel or acknowledge.	M
16	16.6_9	106	88	If the request will be sent to the next hop using a stream-based transport and the copy contains no Content-Length header field, the proxy**MUST** insert one with the correct value for the body of the request (see Section 20.14).	M
16	16.6_10	106	89	A stateful proxy **MUST** create a new client transaction for this request as described in Section 17.1 and instructs the transaction to send the request using the address, port and transport determined in step 7.	M
16	16.6_11	106	90	In order to handle the case where an INVITE request never generates a final response, the TU uses a timer which is called timer C. Timer C**MUST** be set for each client transaction when an INVITE request is proxied.The timer **MUST** be larger than 3 minutes.	M

Section	SubSection	Page	No	Description	Status
16		106	91		M
16	16.7	107	92	When a response is received by an element, it first tries to locate a client transaction (Section 17.1.3) matching the response. If none is found, the element **MUST** process the response (even if it is an informational response) as a stateless proxy (described below).	M
16	16.7	107	93	As client transactions pass responses to the proxy layer, the following processing **MUST** take place: 1. Find the appropriate response context 2. Update timer C for provisional responses 3. Remove the topmost Via 4. Add the response to the response context 5. Check to see if this response should be forwarded immediately	M
16	16.7	107	94	The following processing**MUST** be performed on each response that is forwarded. It is likely that more than one response to each request will be forwarded: at least each provisional and one final response. 7. Aggregate authorization header field values if necessary 8. Optionally rewrite Record-Route header field values 9. Forward the response 10. Generate any necessary CANCEL requests	M
16	16.7_2	108	95	For an INVITE transaction, if the response is a provisional response with status codes 101 to 199 inclusive (i.e., anything but 100), the proxy**MUST** reset timer C for that client transaction.	M
16	16.7_2	108	96	The timer MAY be reset to a different value, but this value**MUST** be greater than 3 minutes.	M
16	16.7_3	108	97	The proxy removes the topmost Via header field value from the response. If no Via header field values remain in the response, the response was meant for this element and **MUST NOT** be forwarded.	M
16	16.7_4	108	98	If the proxy chooses to recurse on any contacts in a 3xx response by adding them to the target set, it **MUST** remove them from the response before adding the response to the response context.	M
16	16.7_4	108	99	However, a proxy **SHOULD NOT** recurse to a non-SIPS URI if the Request-URI of the original request was a SIPS URI.	S

Section	SubSection	Page	No	Description	Status
16	16.7_4	109	100	If the proxy recurses on all of the contacts in a 3xx response, the proxy **SHOULD NOT** add the resulting contactless response to the response context.	S
16	16.7_4	109	101	If a proxy receives a 416 (Unsupported URI Scheme) response to a request whose Request-URI scheme was not SIP, but the scheme in the original received request was SIP or SIPS (that is, the proxy changed the scheme from SIP or SIPS to something else when it proxied a request), the proxy **SHOULD** add a new URI to the target set. This URI **SHOULD** be a SIP URI version of the non-SIP URI that was just tried.	S
16	16.7_4	109	102		S
16	16.7_4	109	103	As with a 3xx response, if a proxy "recurses" on the 416 by trying a SIP or SIPS URI instead, the 416 response **SHOULD NOT** be added to the response context.	S
16	16.7_5	109	104	Until a final response has been sent on the server transaction, the following responses **MUST** be forwarded immediately: - Any provisional response other than 100 (Trying) - Any 2xx response	M
16	16.7_5	109	105	If a 6xx response is received, it is not immediately forwarded, but the stateful proxy **SHOULD** cancel all client pending transactions as described in Section 10, and it **MUST NOT** create any new branches in this context.	S
16			106		M
16	16.7_5	110	107	After a final response has been sent on the server transaction, the following responses **MUST** be forwarded immediately: - Any 2xx response to an INVITE request	M
16	16.7_5	110	108	A stateful proxy **MUST NOT** immediately forward any other responses. In particular, a stateful proxy **MUST NOT** forward any 100 (Trying) response.	M

Section	SubSection	Page	No	Description	Status
16			109		M
16	16.7_5	110	110	Any response chosen for immediate forwarding **MUST** be processed as described in steps "Aggregate Authorization Header Field Values" through "Record-Route".	M
16	16.7_6	110	111	A stateful proxy **MUST** send a final response to a response context's server transaction if no final responses have been immediately forwarded by the above rules and all client transactions in this response context have been terminated.	M
16	16.7_6	110	112	The stateful proxy **MUST** choose the "best" final response among those received and stored in the response context.	M
16	16.7_6	110	113	If there are no final responses in the context, the proxy **MUST** send a 408 (Request Timeout) response to the server transaction.	M
16	16.7_6	110	114	Otherwise, the proxy **MUST** forward a response from the responses stored in the response context. It **MUST** choose from the 6xx class responses if any exist in the context.	M
16	16.7_6	110	115		M
16	16.7_6	110	116	If no 6xx class responses are present, the proxy **SHOULD** choose from the lowest response class stored in the response context.	S
16	16.7_6	111	117	The proxy **SHOULD** give preference to responses that provide information affecting resubmission of this request, such as 401, 407, 415, 420, and 484 if the 4xx class is chosen.	S

Section	SubSection	Page	No	Description	Status
16	16.7_6	111	118	A proxy which receives a 503 (Service Unavailable) response **SHOULD NOT** forward it upstream unless it can determine that any subsequent requests it might proxy will also generate a 503. In other words, forwarding a 503 means that the proxy knows it cannot service any requests, not just the one for the Request-URI in the request which generated the 503.	S
16	16.7_6	111	119	If the only response that was received is a 503, the proxy **SHOULD** generate a 500 response and forward that upstream.	S
16	16.7_6	111	120	The forwarded response **MUST** be processed as described in steps "Aggregate Authorization Header Field Values" through "Record-Route".	M
16	16.7_6	111	121	1xx and 2xx responses may be involved in the establishment of dialogs. When a request does not contain a To tag, the To tag in the response is used by the UAC to distinguish multiple responses to a dialog creating request. A proxy **MUST NOT** insert a tag into the To header field of a 1xx or 2xx response if the request did not contain one. A proxy **MUST NOT** modify the tag in the To header field of a 1xx or 2xx response.	M
16	16.7_6	111	122		M
16	16.7_6	111	123	3-6xx responses are delivered hop-by-hop. When issuing a 3-6xx response, the element is effectively acting as a UAS, issuing its own response, usually based on the responses received from downstream elements. An element **SHOULD** preserve the To tag when simply forwarding a 3-6xx response to a request that did not contain a To tag.	S
16	16.7_6	111	124	A proxy **MUST NOT** modify the To tag in any forwarded response to a request that contains a To tag.	M
16	16.7_7	112	125	If the selected response is a 401 (Unauthorized) or 407 (Proxy Authentication Required), the proxy **MUST** collect any WWW-Authenticate and Proxy-Authenticate header field values from all other 401 (Unauthorized) and 407 (Proxy Authentication Required) responses received so far in this response context and add them to this response without modification before forwarding.	M
16	16.7_8	112	126	If the proxy received the request over TLS, and sent it out over a non-TLS connection, the proxy **MUST** rewrite the URI in the Record-Route header field to be a SIPS URI.	M

Section	SubSection	Page	No	Description	Status
16	16.7_8	112	127	If the proxy received the request over a non-TLS connection, and sent it out over TLS, the proxy **MUST** rewrite the URI in the Record-Route header field to be a SIP URI.	M
16	16.7_8	113	128	The new URI provided by the proxy**MUST** satisfy the same constraints on URIs placed in Record-Route header fields in requests (see Step 4 of Section 16.6) with the following modifications:	M
16	16.7_8	113	129	The URI **SHOULD NOT** contain the transport parameter unless the proxy has knowledge that the next upstream (as opposed to downstream) element that will be in the path of subsequent requests supports that transport.	S
16	16.7_8	113	130	When a proxy does decide to modify the Record-Route header field in the response, one of the operations it performs is locating the Record-Route value that it had inserted. If the request spiraled, and the proxy inserted a Record-Route value in each iteration of the spiral, locating the correct value in the response (which must be the proper iteration in the reverse direction) is tricky. The rules above recommend that a proxy wishing to rewrite Record-Route header field values insert sufficiently distinct URIs into the Record-Route heade	R
16	16.7_9	113	131	The proxy **MUST NOT** add to, modify, or remove the message body.	M
16	16.7_9	113	132	Unless otherwise specified, the proxy**MUST NOT** remove any header field values other than the Via header field value discussed in Section 16.7 Item 3.	M
16	16.7_9	113	133	In particular, the proxy**MUST NOT** remove any "received" parameter it may have added to the next Via header field value while processing the request associated with this response.	M
16	16.7_9	114	134	The proxy **MUST** pass the response to the server transaction associated with the response context. This will result in the response being sent to the location now indicated in the topmost Via header field value.	M
16	16.7_9	114	135	If the server transaction is no longer available to handle the transmission, the element **MUST** forward the response statelessly by sending it to the server transport.	M

Section	SubSection	Page	No	Description	Status
16	16.7_9	114	136	The proxy **MUST** maintain the response context until all of its associated transactions have been terminated, even after forwarding a final response.	M
16	16.7_10	114	137	If the forwarded response was a final response, the proxy**MUST** generate a CANCEL request for all pending client transactions associated with this response context.	M
16	16.7_10	114	138	A proxy **SHOULD** also generate a CANCEL request for all pending client transactions associated with this response context when it receives a 6xx response.	S
16	16.8	114	139	If timer C should fire, the proxy**MUST** either reset the timer with any value it chooses, or terminate the client transaction.	M
16	16.8	114	140	If the client transaction has received a provisional response, the proxy**MUST** generate a CANCEL request matching that transaction.	M
16	16.8	114	141	If the client transaction has not received a provisional response, the proxy **MUST** behave as if the transaction received a 408 (Request Timeout) response.	M
16	16.9	115	142	If the transport layer notifies a proxy of an error when it tries to forward a request (see Section 18.4), the proxy **MUST** behave as if the forwarded request received a 503 (Service Unavailable) response.	M
16	16.9	115	143	If the proxy is notified of an error when forwarding a response, it drops the response. The proxy **SHOULD NOT** cancel any outstanding client transactions associated with this response context due to this notification.	S
16	16.10	115	144	A proxy **MUST** cancel any pending client transactions associated with a response context when it receives a matching CANCEL request.	M

Section	SubSection	Page	No	Description	Status
16	16.10	115	145	While a CANCEL request is handled in a stateful proxy by its own server transaction, a new response context is not created for it. Instead, the proxy layer searches its existing response contexts for the server transaction handling the request associated with this CANCEL. If a matching response context is found, the element **MUST** immediately return a 200 (OK) response to the CANCEL request. In this case, the element is acting as a user agent server as defined in Section 8.2. Furthermore, the element **MUST** generate CANCEL requests for all pending client transactions in the context as described in Section 16.7 step 10.	M
16	16.10	115	146		M
16	16.10	115	147	If a response context is not found, the element does not have any knowledge of the request to apply the CANCEL to. It **MUST** statelessly forward the CANCEL request (it may have statelessly forwarded the associated request previously).	M
16	16.11	116	148	Furthermore, when handling a request statelessly, an element **MUST NOT** generate its own 100 (Trying) or any other provisional response.	M
16	16.11	116	149	A stateless proxy **MUST** validate a request as described in Section 16.3	M
16	16.11	116	150	A stateless proxy **MUST** follow the request processing steps described in Sections 16.4 through 16.5 with the following exception:	M
16	16.11	116	151	o A stateless proxy **MUST** choose one and only one target from the target set. This choice **MUST** only rely on fields in the message and time-invariant properties of the server. In particular, a retransmitted request **MUST** be forwarded to the same destination each time it is processed. Furthermore, CANCEL and non-Routed ACK requests **MUST** generate the same choice as their associated INVITE.	M
16			152		M
16			153		M

Section	SubSection	Page	No	Description	Status
16			154		M
16	16.11	116	155	A stateless proxy **MUST** follow the request processing steps described in Section 16.6 with the following exceptions:	M
16	16.11	116	156	o The requirement for unique branch IDs across space and time applies to stateless proxies as well. However, a stateless proxy cannot simply use a random number generator to compute the first component of the branch ID, as described in Section 16.6 bullet 8. This is because retransmissions of a request need to have the same value, and a stateless proxy cannot tell a retransmission from the original request. Therefore, the component of the branch parameter that makes it unique **MUST** be the same each time a retransmitted request is forwarded. Thus for a stateless proxy, the branch parameter**MUST** be computed as a combinatoric function of message parameters which are invariant on retransmission.	M
16			157		M
16	16.11	117	158	The stateless proxy MAY use any technique it likes to guarantee uniqueness of its branch IDs across transactions. However, the following procedure is **RECOMMENDED**. The proxy examines the branch ID in the topmost Via header field of the received request. If it begins with the magic cookie, the first component of the branch ID of the outgoing request is computed as a hash of the received branch ID. Otherwise, the first component of the branch ID is computed as a hash of the topmost Via, the tag in the To header field, the tag i	R
16	16.11	117	159	o All other message transformations specified in Section 16.6**MUST** result in the same transformation of a retransmitted request. In particular, if the proxy inserts a Record-Route value or pushes URIs into the Route header field, it **MUST** place the same values in retransmissions of the request. As for the Via branch parameter, this implies that the transformations**MUST** be based on time-invariant configuration or retransmission-invariant properties of the request.	M
16			160		M
16			161		M
16	16.11	117	162	Stateless proxies **MUST NOT** perform special processing for CANCEL requests. They are processed by the above rules as any other requests.	M

Section	SubSection	Page	No	Description	Status
16	16.11	118	163	Response processing as described in Section 16.7 does not apply to a proxy behaving statelessly. When a response arrives at a stateless proxy, the proxy **MUST** inspect the sent-by value in the first (topmost) Via header field value. If that address matches the proxy, (it equals a value this proxy has inserted into previous requests) the proxy**MUST** remove that header field value from the response and forward the result to the location indicated in the next Via header field value. The proxy**MUST NOT** add to, modify, or remove the message body. Unless specified otherwise, the proxy**MUST NOT** remove any other header field values. If the address does not match the proxy, the message **MUST** be silently discarded.	M
16			164		M
16			165		M
16			166		M
16			167		M

Section	SubSection	Page	No	Description	Status
17	17.1	125	1	Because of the non-INVITE transaction's reliance on a two-way handshake, TUs **SHOULD** respond immediately to non-INVITE requests.	S
17	17.1.1.2	125	2	The initial state, "calling",**MUST** be entered when the TU initiates a new client transaction with an INVITE request.	M
17	17.1.1.2	125	3	The client transaction**MUST** pass the request to the transport layer for transmission (see Section 18).	M
17	17.1.1.2	125	4	If an unreliable transport is being used, the client transaction**MUST** start timer A with a value of T1.	M
17	17.1.1.2	125	5	If a reliable transport is being used, the client transaction**SHOULD NOT** start timer A (Timer A controls request retransmissions).	S
17	17.1.1.2	125	6	For any transport, the client transaction**MUST** start timer B with a value of 64*T1 seconds (Timer B controls transaction timeouts).	M
17	17.1.1.2	125	7	When timer A fires, the client transaction**MUST** retransmit the request by passing it to the transport layer	M
17	17.1.1.2	125	8	(When timer A fires, the client transaction)**MUST** reset the timer with a value of 2*T1.	M
17	17.1.1.2	126	9	When timer A fires 2*T1 seconds later, the request**MUST** be retransmitted again (assuming the client transaction is still in this state).	M

Section	SubSection	Page	No	Description	Status
17	17.1.1.2	126	10	This process **MUST** continue so that the request is retransmitted with intervals that double after each transmission.	M
17	17.1.1.2	126	11	These retransmissions **SHOULD** only be done while the client transaction is in the "calling" state	S
17	17.1.1.2	126	12	Elements MAY (though it is**NOT RECOMMENDED**) use smaller values of T1 within closed, private networks that do not permit general Internet connection.	R
17	17.1.1.2	126	13	T1 MAY be chosen larger, and this is**RECOMMENDED** if it is known in advance (such as on high latency access links) that the RTT is larger.	R
17	17.1.1.2	126	14	Whatever the value of T1, the exponential backoffs on retransmissions described in this section **MUST** be used.	M
17	17.1.1.2	126	15	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.	S
17	17.1.1.2	126	16	The client transaction **MUST NOT** generate an ACK. The value of 64*T1 is equal to the amount of time required to send seven requests in the case of an unreliable transport.	M
17	17.1.1.2	126	17	In the "Proceeding" state, the client transaction**SHOULD NOT** retransmit the request any longer.	S
17	17.1.1.2	126	18	Furthermore, the provisional response**MUST** be passed to the TU.	M

Section	SubSection	Page	No	Description	Status
17	17.1.1.2	126	19	Any further provisional responses**MUST** be passed up to the TU while in the "Proceeding" state.	M
17	17.1.1.2	126	20	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".	M
17	17.1.1.2	126	21	The client transaction**MUST** pass the received response up to the TU,	M
17	17.1.1.2	126	22	the client transaction**MUST** generate an ACK request, even if the transport is reliable (guidelines for constructing the ACK from the response are given in Section 17.1.1.3) and then pass the ACK to the transport layer for transmission.	M
17	17.1.1.2	126	23	The ACK **MUST** be sent to the same address, port, and transport to which the original request was sent.	M
17	17.1.1.2	126	24	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.	S
17	17.1.1.2	127	25	Any retransmissions of the final response that are received while in the "Completed" state **MUST** cause the ACK to be re-passed to the transport layer for retransmission,	M
17	17.1.1.2	127	26	but the newly received response**MUST NOT** be passed up to the TU.	M
17	17.1.1.2	128	27	If timer D fires while the client transaction is in the "Completed" state, the client transaction **MUST** move to the terminated state.	M

Section	SubSection	Page	No	Description	Status
17	17.1.1.2	128	28	When in either the "Calling" or "Proceeding" states, reception of a 2xx response **MUST** cause the client transaction to enter the "Terminated" state,	M
17	17.1.1.2	128	29	and the response **MUST** be passed up to the TU	M
17	17.1.1.2	129	30	The client transaction **MUST** be destroyed the instant it enters the "Terminated" state.	M
17	17.1.1.3	129	31	A UAC core that generates an ACK for 2xx **MUST** instead follow the rules described in Section 13.	M
17	17.1.1.3	129	32	The ACK request constructed by the client transaction **MUST** contain values for the Call-ID, From, and Request-URI that are equal to the values of those header fields in the request passed to the transport by the client transaction (call this the "original request").	M
17	17.1.1.3	129	33	The To header field in the ACK **MUST** equal the To header field in the response being acknowledged, and therefore will usually differ from the To header field in the original request by the addition of the tag parameter.	M
17	17.1.1.3	129	34	The ACK **MUST** contain a single Via header field,	M
17	17.1.1.3	129	35	and this **MUST** be equal to the top Via header field of the original request. (this = a single Via header field in the ACK)	M
17	17.1.1.3	129	36	The CSeq header field in the ACK **MUST** contain the same value for the sequence number as was present in the original request	M

Section	SubSection	Page	No	Description	Status
17	17.1.1.3	129	37	but the method parameter **MUST** be equal to "ACK".(method = CSeq header field's method)	M
17	17.1.1.3	130	38	If the INVITE request whose response is being acknowledged had Route header fields, those header fields **MUST** appear in the ACK.	M
17	17.1.1.3	130	39	Therefore, placement of bodies in ACK for non-2xx is **NOT RECOMMENDED**, but if done, the body types are restricted to any that appeared in the INVITE, assuming that the response to the INVITE was not 415.	R
17	17.1.2.2	131	40	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*T1 seconds.	S
17	17.1.2.2	131	41	The request **MUST** be passed to the transport layer for transmission.	M
17	17.1.2.2	131	42	If an unreliable transport is in use, the client transaction **MUST** set timer E to fire in T1 seconds.	M
17	17.1.2.2	131	43	If Timer F fires while the client transaction is still in the "Trying" state, the client transaction **SHOULD** inform the TU about the timeout,	S
17	17.1.2.2	131	44	and then it **SHOULD** enter the "Terminated" state.(If Timer F fires while the client transaction is still in the "Trying" state)	S
17	17.1.2.2	131	45	If a provisional response is received while in the "Trying" state, the response **MUST** be passed to the TU,	M

Section	SubSection	Page	No	Description	Status
17	17.1.2.2	131	46	and then the client transaction**SHOULD** move to the "Proceeding" state.(If a provisional response is received while in the "Trying" state,)	S
17	17.1.2.2	131	47	If a final response (status codes 200-699) is received while in the "Trying" state, the response **MUST** be passed to the TU	M
17	17.1.2.2	131	48	and the client transaction**MUST** transition to the "Completed" state.(If a final response (status codes 200-699) is received while in the "Trying" state,)	M
17	17.1.2.2	131	49	If Timer E fires while in the "Proceeding" state, the request**MUST** be passed to the transport layer for retransmission,	M
17	17.1.2.2	131	50	and Timer E **MUST** be reset with a value of T2 seconds.(If Timer E fires while in the "Proceeding" state,)	M
17	17.1.2.2	131	51	If timer F fires while in the "Proceeding" state, the TU**MUST** be informed of a timeout,	M
17	17.1.2.2	131	52	and the client transaction**MUST** transition to the terminated state.	M
17	17.1.2.2	131	53	If a final response (status codes 200-699) is received while in the "Proceeding" state, the response **MUST** be passed to the TU	M
17	17.1.2.2	131	54	and the client transaction**MUST** transition to the "Completed" state.(If a final response (status codes 200-699) is received while in the "Proceeding" state,)	M

Section	SubSection	Page	No	Description	Status
17	17.1.2.2	131	55	Once the client transaction enters the "Completed" state, it**MUST** set Timer K to fire in T4 seconds for unreliable transports, and zero seconds for reliable transports.	M
17	17.1.2.2	132	56	If Timer K fires while in this state, the client transactio**MUST** transition to the "Terminated" state.(this state="Completed")	M
17	17.1.2.2	132	57	Once the transaction is in the terminated state, it**MUST** be destroyed immediately.	M
17	17.1.3	134	58	The client transaction**SHOULD** inform the TU that a transport failure has occurred,	S
17	17.1.3	134	59	and the client transaction**SHOULD** transition directly to the "Terminated" state.	S
17	17.2.1	134	60	The server transaction **MUST** generate a 100 (Trying) response unless it knows that the TU will generate a provisional or final response within 200 ms, in which case it MAY generate a 100 (Trying) response.	M
17	17.2.1	134	61	The 100 (Trying) response is constructed according to the procedures in Section 8.2.6, except that the insertion of tags in the To header field of the response (when none was present in the request) is downgraded from MAY to **SHOULD NOT**.	S
17	17.2.1	134	62	The request **MUST** be passed to the TU.	M
17	17.2.1	134	63	The TU passes any number of provisional responses to the server transaction. So long as the server transaction is in the "Proceeding" state, each of these **MUST** be passed to the transport layer for transmission.	M

Section	SubSection	Page	No	Description	Status
17	17.2.1	134	64	If a request retransmission is received while in the "Proceeding" state, the most recent provisional response that was received from the TU**MUST** be passed to the transport layer for retransmission.	M
17	17.2.1	134	65	If, while in the "Proceeding" state, the TU passes a 2xx response to the server transaction, the server transaction **MUST** pass this response to the transport layer for transmission.	M
17	17.2.1	135	66	The server transaction **MUST** then transition to the "Terminated" state.	M
17	17.2.1	135	67	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,	M
17	17.2.1	135	68	the state machine **MUST** enter the "Completed" state.(While in the "Proceeding" state, if the TU passes a response with status code from 300 to 699 to the server transaction,)	M
17	17.2.1	135	69	When the "Completed" state is entered, timer H **MUST** be set to fire in 64*T1 seconds for all transports.	M
17	17.2.1	135	70	Furthermore, while in the "Completed" state, if a request retransmission is received, the server **SHOULD** pass the response to the transport for retransmission.	S
17	17.2.1	135	71	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.	M
17	17.2.1	135	72	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	M

Section	SubSection	Page	No	Description	Status
17	17.2.1	135	73	(If timer H fires while in the "Completed" state, it implies that the ACK was never received. In this case, the server transaction) **MUST** indicate to the TU that a transaction failure has occurred.	M
17	17.2.1	137	74	Once timer I fires, the server **MUST** transition to the "Terminated" state.	M
17	17.2.1	137	75	Once the transaction is in the "Terminated" state, it **MUST** be destroyed immediately. As with client transactions, this is needed to ensure reliability of the 2xx responses to INVITE.	M
17	17.2.2	137	76	While in the "Trying" state, if the TU passes a provisional response to the server transaction, the server transaction **MUST** enter the "Proceeding" state.	M
17	17.2.2	137	77	The response **MUST** be passed to the transport layer for transmission.	M
17	17.2.2	137	78	Any further provisional responses that are received from the TU while in the "Proceeding" state **MUST** be passed to the transport layer for transmission.	M
17	17.2.2	137	79	If a retransmission of the request is received while in the "Proceeding" state, the most recently sent provisional response **MUST** be passed to the transport layer for retransmission.	M
17	17.2.2	137	80	If the TU passes a final response (status codes 200-699) to the server while in the "Proceeding" state, the transaction **MUST** enter the "Completed" state,	M
17	17.2.2	137	81	and the response **MUST** be passed to the transport layer for transmission.(If the TU passes a final response (status codes 200-699) to the server while in the "Proceeding" state,)	M

Section	SubSection	Page	No	Description	Status
17	17.2.2	137	82	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.	M
17	17.2.2	137	83	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.	M
17	17.2.2	137	84	Any other final responses passed by the TU to the server transaction**MUST** be discarded while in the "Completed" state.	M
17	17.2.2	137	85	The server transaction remains in this state until Timer J fires, at which point it **MUST** transition to the "Terminated" state.	M
17	17.2.2	137	86	The server transaction **MUST** be destroyed the instant it enters the "Terminated" state. 17.2.3 Matching Requests to Server Transactions	M
17	17.2.4	141	87	First, the procedures in [4] are followed, which attempt to deliver the response to a backupIf those should all fail, based on the definition of failure in [4], the server transaction **SHOULD** inform the TU that a failure has occurred,	S
17	17.2.4	141	88	**SHOULD** transition to the terminated state.(First, the procedures in [4] are followed, which attempt to deliver the response to a backup)	S

Section	SubSection	Page	No	Description	Status
18		142	1	It is **RECOMMENDED** that connections be kept open for some implementation-defined duration after the last message was sent or received over that connection.	R
18		142	2	This duration **SHOULD** at least equal the longest amount of time the element would need in order to bring a transaction from instantiation to the terminated state.	S
18		142	3	All SIP elements **MUST** implement UDP and TCP. SIP elements MAY implement other protocols.	M
18	1	142	4	Making TCP mandatory for the UA is a substantial change from RFC 2543. It has arisen out of the need to handle larger messages, which**MUST** use TCP, as discussed below.Thus, even if an element never sends large messages, it may receive one and needs to be able to handle them.	M
18	2	142	5	If a request is within 200 bytes of the path MTU, or if it is larger than 1300 bytes and the path MTU is unknown, the request**MUST** be sent using an RFC 2914 [43] congestion controlled transport protocol, such as TCP.	M
18	18.1.1	142	6	If this causes a change in the transport protocol from the one indicated in the top Via, the value in the top Via**MUST** be changed.	M
18	18.1.1	142	7	This prevents fragmentation of messages over UDP and provides congestion control for larger messages. However, implementations **MUST** be able to handle messages up to the maximum datagram packet size.	M
18	18.1.1	143	8	If an element sends a request over TCP because of these message size constraints, and that request would have otherwise been sent over UDP, if the attempt to establish the connection generates either an ICMP Protocol Not Supported, or results in a TCP reset, the element**SHOULD** retry the request, using UDP.	S
18	18.1.1	143	9	A client that sends a request to a multicast address**MUST** add the "maddr" parameter to its Via header field value containing the destination multicast address,	M

Section	SubSection	Page	No	Description	Status
18	18.1.1	143	10	for IPv4, **SHOULD** add the "ttl" parameter with a value of 1. Usage of IPv6 multicast is not defined in this specification, and will be a subject of future standardization when the need arises.	S
18	18.1.1	143	11	Before a request is sent, the client transport**MUST** insert a value of the "sent-by" field into the Via header field.	M
18	18.1.1	143	12	The usage of an FQDN is **RECOMMENDED**.(The=Via)	R
18	18.1.1	143	13	Therefore, the client transport **MUST** be prepared to receive the response on the same connection used to send the request. Under error conditions, the server may attempt to open a new connection to send the response.	M
18	18.1.1	143	14	To handle this case, the transport layer**MUST** also be prepared to receive an incoming connection on the source IP address from which the request was sent and port number in the "sent-by" field.	M
18	18.1.1	144	15	It also **MUST** be prepared to receive incoming connections on any address and port that would be selected by a server based on the procedures described in Section 5 of [4].	M
18	18.1.1	144	16	For unreliable unicast transports, the client transport**MUST** be prepared to receive responses on the source IP address from which the request is sent (as responses are sent back to the source address) and the port number in the "sent-by" field.	M
18	18.1.1	144	17	The client **MUST** be prepared to receive responses on any address and port that would be selected by a server based on the procedures described in Section 5 of [4].	M
18	18.1.1	144	18	For multicast, the client transport **MUST** be prepared to receive responses on the same multicast group and port to which the request is sent (that is, it needs to be a member of the multicast group it sent the request to.)	M

Section	SubSection	Page	No	Description	Status
18	18.1.1	144	19	If a request is destined to an IP address, port, and transport to which an existing connection is open, it is**RECOMMENDED** that this connection be used to send the request, but another connection MAY be opened and used.	M
18	18.1.2	144	20	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.	M
18	18.1.2	144	21	If there are any client transactions in existence, the client transport uses the matching procedures of Section 17.1.3 to attempt to match the response to an existing transaction. If there is a match, the response**MUST** be passed to that transaction.	M
18	18.1.2	144	22	(If there are any client transactions in existence, the client transport uses the matching procedures of Section 17.1.3 to attempt to match the response to an existing transaction. If there is a match, the response MUST be passed to that transaction.)Otherwise, the response**MUST** be passed to the core (whether it be stateless proxy, stateful proxy, or UA) for further processing. Handling of these "stray" responses is dependent on the core (a proxy will forward them, while a UA will discard, for example). 18.2 Server:	M
18	18.2.1	145	23	A server **SHOULD** be prepared to receive requests on any IP address, port and transport combination that can be the result of a DNS lookup on a SIP or SIPS URI [4] that is handed out for the purposes of communicating with that server.	S
18	18.2.1	145	24	It is also **RECOMMENDED** that a server listen for requests on the default SIP ports (5060 for TCP and UDP, 5061 for TLS over TCP) on all public interfaces.	R
18	18.2.1	145	25	For any port and interface that a server listens on for UDP, it**MUST** listen on that same port and interface for TCP.	M
18	18.2.1	145	26	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.	M
18	18.2.1	145	27	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.	M

Section	SubSection	Page	No	Description	Status
18	18.2.1	145	28	This parameter **MUST** contain the source address from which the packet was received.	M
18	18.2.2	146	29	The server transport uses the value of the top Via header field in order to determine where to send a response. It **MUST** follow the following process:	M
18	18.2.2	146	30	o If the "sent-protocol" is a reliable transport protocol such as TCP or SCTP, or TLS over those, the response **MUST** be sent using the existing connection to the source of the original request that created the transaction, if that connection is still open.	M
18	18.2.2	146	31	This requires the server transport to maintain an association between server transactions and transport connections. If that connection is no longer open, the server **SHOULD** open a connection to the IP address in the "received" parameter, if present, using the port in the "sent-by" value, or the default port for that transport, if no port is specified.	S
18	18.2.2	146	32	If that connection attempt fails, the server **SHOULD** use the procedures in [4] for servers in order to determine the IP address and port to open the connection and send the response to.	S
18	18.2.2	146	33	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.	M
18	18.2.2	146	34	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.	S
18	18.2.2	146	35	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.	M
18	18.2.2	146	36	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.	S

Section	SubSection	Page	No	Description	Status
18	18.2.2	147	37	o Otherwise, if it is not receiver-tagged, the responseMUST be sent to the address indicated by the "sent-by" value, using the procedures in Section 5 of [4].	M
18	18.3	147	38	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.	M
18	18.3	147	39	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.	M
18	18.3	147	40	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.	S
18	18.3	147	41	In the case of stream-oriented transports such as TCP, the Content- Length header field indicates the size of the body. The Content- Length header field MUST be used with stream oriented transports	M
18	18.4	147	42	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.	S
18	18.4	147	43	Source quench and TTL exceeded ICMP errorsSHOULD be ignored.	S
18	18.4	147	44	If the transport user asks for a request to be sent over a reliable transport, and the result is a connection failure, the transport layerSHOULD inform the transport user of a failure in sending.	S

Section	SubSection	Page	No	Description	Status
19	19.1.1	149	1	If the @ sign is present in a SIP or SIPS URI, the user field **MUST NOT** be empty	M
19	19.1.1	149	2	password: A password associated with the user. While the SIP and SIPS URI syntax allows this field to be present, its use is **NOT RECOMMENDED**, because the passing of authentication information in clear text (such as URIs) has proven to be a security risk in almost every case where it has been used. For instance, transporting a PIN number in this field exposes the PIN.	R
19	19.1.1	149	3	host: The host providing the SIP resource. The host part contains either a fully-qualified domain name or numeric IPv4 or IPv6 address. Using the fully-qualified domain name form is **RECOMMENDED** whenever possible.	R
19	19.1.1	149	4	Even though an arbitrary number of URI parameters may be included in a URI, any given parameter-name **MUST NOT** appear more than once.	M
19	19.1.1	150	5	For a SIPS URI, the transport parameter **MUST** indicate a reliable transport.	M
19	19.1.1	150	6	The ttl parameter determines the time-to-live value of the UDP multicast packet and **MUST** only be used if maddr is a multicast address and the transport protocol is UDP. For example, to specify a call to alice@atlanta.com using multicast to 239.255.255.1 with a ttl of 15, the following URI would be used:	M
19	19.1.1	150	7	If the user string contains a telephone number formatted as a telephone-subscriber, the user parameter value "phone" **SHOULD** be present. Even without this parameter, recipients of SIP and SIPS URIs MAY interpret the pre-@ part as a telephone number if local restrictions on the name space for user name allow it.	S
19	19.1.1	151	8	Since the uri-parameter mechanism is extensible, SIP elements **MUST** silently ignore any uri-parameters that they do not understand.	M
19	19.1.1	151	9	The external column describes URIs appearing anywhere outside of a SIP message, for instance on a web page or business card. Entries marked "m" are mandatory, those marked "o" are optional, and those marked "-" are not allowed. Elements processing URIs **SHOULD** ignore any disallowed components if they are present.	S

Section	SubSection	Page	No	Description	Status
19	19.1.2	152	10	Excluded US- ASCII characters (RFC 2396 [5]), such as space and control characters and characters used as URI delimiters, also**MUST** be escaped.	M
19	19.1.2	152	11	URIs **MUST NOT** contain unescaped space and control characters.	M
19	19.1.2	152	12	For each component, the set of valid BNF expansions defines exactly which characters may appear unescaped. All other characters**MUST** be escaped.	M
19	19.1.2	153	13	Expanding the hname and hvalue tokens in Section 25 show that all URI reserved characters in header field names and values**MUST** be escaped.	M
19	19.1.2	153	14	Any characters occurring in a telephone-subscriber that do not appear in an expansion of the BNF for the user rule**MUST** be escaped	M
19	19.1.2	153	15	Current implementations **MUST NOT** attempt to improve robustness by treating received escaped characters in the host component as literally equivalent to their unescaped counterpart.	M
19	19.1.4	155	16	- All other uri-parameters appearing in only one URI are ignored when comparing the URIs. o URI header components are never ignored. Any present header component **MUST** be present in both URIs and match for the URIs to match. The matching rules are defined for each header field in Section 20.	M
19	19.1.5	156	17	An implementation **MUST** include any provided transport, maddr, ttl, or user parameter in the Request-URI of the formed request.	M
19	19.1.5	156	18	If the URI contains a method parameter, its value**MUST** be used as the method of the request.	M

Section	SubSection	Page	No	Description	Status
19	19.1.5	156	19	The method parameter **MUST NOT** be placed in the Request-URI.	M
19	19.1.5	156	20	Unknown URI parameters **MUST** be placed in the message's Request-URI	M
19	19.1.5	156	21	An implementation **SHOULD** treat the presence of any headers or body parts in the URI as a desire to include them in the message, and choose to honor the request on a per-component basis.	S
19	19.1.5	156	22	An implementation **SHOULD NOT** honor these obviously dangerous header fields: From, Call-ID, CSeq, Via, and Record-Route.	S
19	19.1.5	156	23	An implementation **SHOULD NOT** honor any requested Route header field values in order to not be used as an unwitting agent in malicious attacks.	S
19	19.1.5	156	24	An implementation **SHOULD NOT** honor requests to include header fields that may cause it to falsely advertise its location or capabilities. These include: Accept, Accept-Encoding, Accept-Language, Allow, Contact (in its dialog usage), Organization, Supported, and User- Agent.	S
19	19.1.5	156	25	An implementation **SHOULD** verify the accuracy of any requested descriptive header fields, including: Content-Disposition, Content- Encoding, Content-Language, Content-Length, Content-Type, Date, Mime-Version, and Timestamp.	S
19	19.1.5	156	26	If the request formed from constructing a message from a given URI is not a valid SIP request, the URI is invalid. An implementation**MUST NOT** proceed with transmitting the request. It should instead pursue the course of action due an invalid URI in the context it occurs.	M
19	19.1.5	157	27	The URI might indicate use of an unimplemented transport or extension, for example. An implementation **SHOULD** refuse to send these requests rather than modifying them to match their capabilities.	S

Section	SubSection	Page	No	Description	Status
19	19.1.5	157	28	An implementation **MUST NOT** send a request requiring an extension that it does not support.	M
19	19.1.6	158	29	To mitigate this problem, elements constructing telephone-subscriber fields to place in the userinfo part of a SIP or SIPS URI**SHOULD** fold any case-insensitive portion of telephone-subscriber to lower case, and order the telephone-subscriber parameters lexically by parameter name, excepting isdn-subaddress and post-dial, which occur first and in that order. (All components of a tel URL except for future- extension parameters are defined to be compared case-insensitive.)(this problem = P158 ex	S
19	19.3	159	30	When a tag is generated by a UA for insertion into a request or response, it **MUST** be globally unique and cryptographically random with at least 32 bits of randomness.	M

Section	SubSection	Page	No	Description	Status
20		160	1	m*: The header field **SHOULD** be sent, but clients/servers need to be prepared to receive messages without that header field.	S
20		160	2	t: The header field **SHOULD** be sent, but clients/servers need to be prepared to receive messages without that header field.	S
20		160	3	If a stream-based protocol (such as TCP) is used as a transport, then the header field **MUST** be sent. *: The header field is required if the message body is not empty. See Sections 20.14, 20.15 and 7.4 for details.	M
20		161	4	A "mandatory" header field **MUST** be present in a request,	M
20		161	5	(A "mandatory" header field) **MUST** be understood by the UAS receiving the request.	M
20		161	6	A mandatory response header field **MUST** be present in the response,	M
20		161	7	the header field **MUST** be understood by the UAC processing the response.	M
20		161	8	"Not applicable" means that the header field **MUST NOT** be present in a request.	M
20		161	9	If one is placed in a request by mistake, it **MUST** be ignored by the UAS receiving the request.	M

Section	SubSection	Page	No	Description	Status
20		161	10	Similarly, a header field labeled "not applicable" for a response means that the UAS **MUST NOT** place the header field in the response,	M
20		161	11	and the UAC **MUST** ignore the header field in the response.	M
20		161	12	A UA **SHOULD** ignore extension header parameters that are not understood.	S
20		161	13	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 >).	M
20	20.1	161	14	If the URI contains a comma, question mark or semicolon, the URI**MUST** be enclosed in angle brackets (< and >).	S
20	20.2	163	15	If no Accept-Encoding header field is present, the server**SHOULD** assume a default value of identity. This differs slightly from the HTTP definition, which indicates that when not present, any encoding can be used, but the identity encoding is preferred.	S
20	20.3	164	16	The Accept-Language header field is used in requests to indicate the preferred languages for reason phrases, session descriptions, or status responses carried as message bodies in the response. If no Accept-Language header field is present, the server **SHOULD** assume all languages are acceptable to the client.	S
20	20.4	164	17	In addition, a user **SHOULD** be able to disable this feature selectively.	S
20	20.5	165	18	All methods, including ACK and CANCEL, understood by the UA**MUST** be included in the list of methods in the Allow header field, when present.	M

Section	SubSection	Page	No	Description	Status
20	20.5	165	19	The absence of an Allow header field**MUST NOT** be interpreted to mean that the UA sending the message supports no methods. Rather, it implies that the UA is not providing any information on what methods it supports.	M
20	20.7	165	20	Although not a comma- separated list, this header field name may be present multiple times, and **MUST NOT** be combined into a single header line using the usual rules described in Section 7.3.	M
20	20.9	166	21	Use of the Call-Info header field can pose a security risk. If a callee fetches the URIs provided by a malicious caller, the callee may be at risk for displaying inappropriate or offensive content, dangerous or illegal content, and so on. Therefore, it is **RECOMMENDED** that a UA only render the information in the Call-Info header field if it can verify the authenticity of the element that originated the header field and trusts that element.	R
20	20.10	167	22	Even if the "display-name" is empty, the "name-addr" form **MUST** be used if the "addr-spec" contains a comma, semicolon, or question mark. There may or may not be LWS between the display-name and the "<".	M
20	20.11	168	23	For backward-compatibility, if the Content-Disposition header field is missing, the server **SHOULD** assume bodies of Content-Type application/sdp are the disposition "session", while other content types are "render".	S
20	20.11	168	24	The handling parameter, handling-param, describes how the UAS should react if it receives a message body whose content type or disposition type it does not understand. The parameter has defined values of "optional" and "required". If the handling parameter is missing, the value "required"**SHOULD** be assumed.	S
20	20.12	169	25	When present, its value indicates what additional content codings have been applied to the entity-body, and thus what decoding mechanisms**MUST** be applied in order to obtain the media-type referenced by the Content-Type header field.	M
20	20.12	169	26	If multiple encodings have been applied to an entity-body, the content codings **MUST** be listed in the order in which they were applied.	M
20	20.12	169	27	Clients MAY apply content encodings to the body in requests. A server MAY apply content encodings to the bodies in responses. The server**MUST** only use encodings listed in the Accept-Encoding header field in the request.	M

Section	SubSection	Page	No	Description	Status
20	20.14	169	28	Applications **SHOULD** use this field to indicate the size of the message-body to be transferred, regardless of the media type of the entity.	S
20	20.14	169	29	If a stream-based protocol (such as TCP) is used as transport, the header field **MUST** be used.	M
20	20.14	169	30	Any Content-Length greater than or equal to zero is a valid value. If no body is present in a message, then the Content-Length header field value**MUST** be set to zero.	M
20	20.15	170	31	The "media-type" element is defined in [H3.7]. The Content-Type header field **MUST** be present if the body is not empty.	M
20	20.16	170	32	The sequence number **MUST** be expressible as a 32-bit unsigned integer.	M
20	20.20	172	33	A system **SHOULD** use the display name "Anonymous" if the identity of the client is to remain hidden.	S
20	20.20	172	34	Even if the "display- name" is empty, the "name-addr" form **MUST** be used if the "addr-spec" contains a comma, question mark, or semicolon.	M
20	20.26	174	35	The Priority header field indicates the urgency of the request as perceived by the client. The Priority header field describes the priority that the SIP request should have to the receiving human or its agent. For example, it may be factored into decisions about call routing and acceptance. For these decisions, a message containing no Priority header field**SHOULD** be treated as if it specified a Priority of "normal".	S
20	20.26	174	36	It is **RECOMMENDED** that the value of "emergency" only be used when life, limb, or property are in imminent danger.	R

Section	SubSection	Page	No	Description	Status
20	20.28	175	37	Although not a comma-separated list, this header field name may be present multiple times, and **MUST NOT** be combined into a single header line using the usual rules described in Section 7.3.1.	M
20	20.31	176	38	If the user wished to remain anonymous, the header field**SHOULD** either be omitted from the request or populated in such a way that does not reveal any private information.	S
20	20.31	176	39	Even if the "display-name" is empty, the "name-addr" form **MUST** be used if the "addr-spec" contains a comma, question mark, or semicolon. Syntax issues are discussed in Section 7.3.1.	M
20	20.32	176	40	The Require header field is used by UACs to tell UASs about options that the UAC expects the UAS to support in order to process the request. Although an optional header field, the Require**MUST NOT** be ignored if it is present.	M
20	20.32	176	41	The Require header field contains a list of option tags, described in Section 19.2. Each option tag defines a SIP extension that**MUST** be understood to process the request.	M
20	20.32	176	42	A UAC compliant to this specification**MUST** only include option tags corresponding to standards-track RFCs.	M
20	20.35	177	43	Revealing the specific software version of the server might allow the server to become more vulnerable to attacks against software that is known to contain security holes. Implementers **SHOULD** make the Server header field a configurable option.	S
20	20.37	178	44	The Supported header field contains a list of option tags, described in Section 19.2, that are understood by the UAC or UAS. A UA compliant to this specification **MUST** only include option tags corresponding to standards-track RFCs. If empty, it means that no extensions are supported.	M
20	20.41	179	45	Revealing the specific software version of the user agent might allow the user agent to become more vulnerable to attacks against software that is known to contain security holes. Implementers**SHOULD** make the User-Agent header field a configurable option.	S

Section	SubSection	Page	No	Description	Status
20	20.42	180	46	For implementations compliant to this specification, the value of the branch parameter **MUST** start with the magic cookie "z9hG4bK", as discussed in Section 8.1.1.7.	M
20	20.43	182	47	399 Miscellaneous warning: The warning text can include arbitrary information to be presented to a human user or logged. A system receiving this warning**MUST NOT** take any automated action.	M

Section	SubSection	Page	No	Description	Status
21		182	1	Not all HTTP/1.1 response codes are appropriate, and only those that are appropriate are given here. Other HTTP/1.1 response codes**SHOULD NOT** be used.	S
21	21.3.1	184	2	The choices **SHOULD** also be listed as Contact fields (Section 20.10).	S
21	21.3.2	184	3	The user can no longer be found at the address in the Request-URI, and the requesting client **SHOULD** retry at the new address given by the Contact header field (Section 20.10).	S
21	21.3.2	184	4	The requestor **SHOULD** update any local directories, address books, and user location caches with this new value and redirect future requests to the address(es) listed.	S
	21.3.3	184	5	The requesting client**SHOULD** retry the request at the new address(es) given by the Contact header field (Section 20.10). The Request-URI of the new request uses the value of the Contact header field in the response.	S
21	21.3.3	185	6	The duration of the validity of the Contact URI can be indicated through an Expires (Section 20.19) header field or an expires parameter in the Contact header field. Both proxies and UAs MAY cache this URI for the duration of the expiration time. If there is no explicit expiration time, the address is only valid once for recursing, and **MUST NOT** be cached for future	M
21	21.3.4	185	7	The requested resource **MUST** be accessed through the proxy given by the Contact field. The Contact field gives the URI of the proxy.	M
21	21.3.4	185	8	The recipient is expected to repeat this single request via the proxy. 305 (Use Proxy) responses **MUST** only be generated by UASs.	M
21	21.4	185	9	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).	S

Section	SubSection	Page	No	Description	Status
21	21.4.1	185	10	The request could not be understood due to malformed syntax. The Reason-Phrase **SHOULD** identify the syntax problem in more detail, for example, "Missing Call-ID header field".	S
21	21.4.4	186	11	The server understood the request, but is refusing to fulfill it. Authorization will not help, and the request **SHOULD NOT** be repeated.	S
21	21.4.6	186	12	The response **MUST** include an Allow header field containing a list of valid methods for the indicated address.	M
21	21.4.8	186	13	This code is similar to 401 (Unauthorized), but indicates that the clien**MUST** first authenticate itself with the proxy. SIP access authentication is explained in Sections 26 and 22.3.	M
21	21.4.10	187	14	The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found)**SHOULD** be used instead.	S
21	21.4.11	187	15	If the condition is temporary, the server**SHOULD** include a Retry- After header field to indicate that it is temporary and after what time the client MAY try again.	S
21	21.4.13	187	16	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.	M
21	21.4.15	187	17	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. 21.4.16 421 Extension Required	M
21	21.4.16	188	18	The UAS needs a particular extension to process the request, but this extension is not listed in a Supported header field in the request. Responses with this status code **MUST** contain a Require header field listing the required extensions.	M

Section	SubSection	Page	No	Description	Status
21	21.4.16	188	19	A UAS **SHOULD NOT** use this response unless it truly cannot provide any useful service to the client.	M
21	21.4.16	188	20	Instead, if a desirable extension is not listed in the Supported header field, servers **SHOULD** process the request using baseline SIP capabilities and any extensions supported by the client.	S
21	21.4.18	188	21	The user could also be available elsewhere (unbeknownst to this server). The reason phrase **SHOULD** indicate a more precise cause as to why the callee is unavailable.	S
21	21.4.18	188	22	This value **SHOULD** be settable by the UA. Status 486 (Busy Here) MAY be used to more precisely indicate a particular reason for the call failure.	S
21	21.4.22	189	23	The server received a request with a Request-URI that was incomplete. Additional information **SHOULD** be provided in the reason phrase.	S
21	21.4.23	189	24	Revealing alternatives can infringe on privacy of the user or the organization. It **MUST** be possible to configure a server to respond with status 404 (Not Found) or to suppress the listing of possible choices for ambiguous Request-URIs.	M
21	21.4.24	190	25	Status 600 (Busy Everywhere) **SHOULD** be used if the client knows that no other end system will be able to accept this call.	S
21	21.5.4	191	26	The server is temporarily unable to process the request due to a temporary overloading or maintenance of the server. The server MAY indicate when the client should retry the request in a Retry-After header field. If no Retry-After is given, the client **MUST** act as if it had received a 500 (Server Internal Error) response.	M
21	21.5.4	191	27	A client (proxy or UAC) receiving a 503 (Service Unavailable) **SHOULD** attempt to forward the request to an alternate server.	S

Section	SubSection	Page	No	Description	Status
21	21.5.4	191	28	It **SHOULD NOT** forward any other requests to that server for the duration specified in the Retry-After header field, if present.	S

Section	SubSection	Page	No	Description	Status
22		193	1	Once the originator has been identified, the recipient of the request **SHOULD** ascertain whether or not this user is authorized to make the request in question.	S
22		193	2	Servers **MUST NOT** accept credentials using the "Basic" authorization scheme, and servers also **MUST NOT** challenge with "Basic".	M
			3		M
22	22.1	193	4	In SIP, a UAS uses the 401 (Unauthorized) response to challenge the identity of a UAC. Additionally, registrars and redirect servers MAY make use of 401 (Unauthorized) responses for authentication, but proxies **MUST NOT**, and instead MAY use the 407 (Proxy Authentication Required) response. The requirements for inclusion of the Proxy-Authenticate, Proxy-Authorization, WWW-Authenticate, and Authorization in the various messages are identical to those described in RFC 2617 [17].	M
22	22.1	194	5	Operators of user agents or proxy servers that will authenticate received requests **MUST** adhere to the following guidelines for creation of a realm string for their server:	M
22	22.1	194	6	(Operators of user agents or proxy servers that will authenticate received requests MUST adhere to the following guidelines for creation of a realm string for their server:) o Realm strings **MUST** be globally unique.	M
22	22.1	194	7	(Operators of user agents or proxy servers that will authenticate received requests MUST adhere to the following guidelines for creation of a realm string for their server:) It is **RECOMMENDED** that a realm string contain a hostname or domain name, following the recommendation in Section 3.2.1 of RFC 2617 [17].	R
22	22.1	194	8	(Operators of user agents or proxy servers that will authenticate received requests MUST adhere to the following guidelines for creation of a realm string for their server:) o Realm strings **SHOULD** present a human-readable identifier that can be rendered to a user.	S
22	22.1	195	9	For this reason, any credentials in the INVITE that were accepted by a server **MUST** be accepted by that server for the ACK.	M

Section	SubSection	Page	No	Description	Status
22	22.1	195	10	UACs creating an ACK message will duplicate all of the Authorization and Proxy-Authorization header field values that appeared in the INVITE to which the ACK corresponds. Servers **MUST NOT** attempt to challenge an ACK.	M
22	22.1	195	11	Although the CANCEL method does take a response (a 2xx), servers**MUST NOT** attempt to challenge CANCEL requests since these requests cannot be resubmitted.	M
22	22.1	195	12	Generally, a CANCEL request **SHOULD** be accepted by a server if it comes from the same hop that sent the request being canceled (provided that some sort of transport or network layer security association, as described in Section 26.2.1, is in place).	S
22	22.1	195	13	When a UAC receives a challenge, it**SHOULD** render to the user the contents of the "realm" parameter in the challenge (which appears in either a WWW-Authenticate header field or Proxy-Authenticate header field) if the UAC device does not already know of a credential for the realm in question.	S
22	22.1	195	14	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).	M
22	22.2	195	15	The WWW-Authenticate response-header field **MUST** be included in 401 (Unauthorized) response messages.	M
22	22.2	196	16	When the originating UAC receives the 401 (Unauthorized), i**SHOULD**, if it is able, re-originate the request with the proper credentials.	S
22	22.2	196	17	Once authentication credentials have been supplied (either directly by the user, or discovered in an internal keyring), UAs**SHOULD** cache the credentials for a given value of the To header field and "realm" and attempt to re-use these values on the next request for that destination.	S
22	22.2	196	18	When a UAC resubmits a request with its credentials after receiving a 401 (Unauthorized) or 407 (Proxy Authentication Required) response, i**MUST** increment the CSeq header field value as it would normally when sending an updated request.	M

Section	SubSection	Page	No	Description	Status
22	22.3	197	19	The proxy **MUST** populate the 407 (Proxy Authentication Required) message with a Proxy- Authenticate header field value applicable to the proxy for the requested resource.	M
22	22.3	197	20	The use of Proxy-Authenticate and Proxy-Authorization parallel that described in [17], with one difference. Proxies**MUST NOT** add values to the Proxy-Authorization header field.	M
22	22.3	197	21	All 407 (Proxy Authentication Required) responses**MUST** be forwarded upstream toward the UAC following the procedures for any other response.	M
22	22.3	197	22	When the originating UAC receives the 407 (Proxy Authentication Required) it **SHOULD**, if it is able, re-originate the request with the proper credentials.	S
22	22.3	197	23	The UAC **SHOULD** also cache the credentials used in the re-originated request.	S
22	22.3	197	24	The following rule is**RECOMMENDED** for proxy credential caching:	R
22	22.3	197	25	These credentials **MUST NOT** be cached across dialogs; however, if a UA is configured with the realm of its local outbound proxy, when one exists, then the UA MAY cache credentials for that realm across dialogs.	M
22	22.3	198	26	When multiple proxies are used in a chain, a Proxy- Authorization header field value **MUST NOT** be consumed by any proxy whose realm does not match the "realm" parameter specified in that value.	M
22	22.3	198	27	Note that if an authentication scheme that does not support realms is used in the Proxy-Authorization header field, a proxy serve**MUST** attempt to parse all Proxy-Authorization header field values to determine whether one of them has what the proxy server considers to be valid credentials.	M

Section	SubSection	Page	No	Description	Status
22	22.3	198	28	Because this is potentially very time- consuming in large networks, proxy servers **SHOULD** use an authentication scheme that supports realms in the Proxy-Authorization header field.	S
22	22.3	198	29	Each WWW-Authenticate and Proxy-Authenticate value received in responses to the forked request **MUST** be placed into the single response that is sent by the forking proxy to the UA; the ordering of these header field values is not significant.	M
22	22.3	199	30	As noted above, multiple credentials in a request**SHOULD** be differentiated by the "realm" parameter.	S
22	22.3	199	31	When it retries a request, a UAC MAY therefore supply multiple credentials in Authorization or Proxy-Authorization header fields with the same "realm" parameter value. The same credentials **SHOULD** be used for the same realm.	S
22	22.4	199	32	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.	M
22	22.4	199	33	Note, however, that SIP servers **MUST NOT** accept or request Basic authentication.	M
22	22.4	199	34	(The rules for Digest authentication follow those defined in [17], with "HTTP/1.1" replaced by "SIP/2.0" in addition to the following differences:) 2. The BNF in RFC 2617 has an error in that the 'uri' parameter of the Authorization header field for HTTP Digest authentication is not enclosed in quotation marks. (The example in Section 3.5 of RFC 2617 is correct.) For SIP, the 'uri' **MUST** be enclosed in quotation marks.	M
22	22.4	200	35	(The rules for Digest authentication follow those defined in [17], with "HTTP/1.1" replaced by "SIP/2.0" in addition to the following differences:) 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.	M
22	22.4	200	36	However, servers **MUST** always send a "qop" parameter in WWW-Authenticate and Proxy-Authenticate header field values.	M

Section	SubSection	Page	No	Description	Status
22	22.4	200	37	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).	M
22	22.4	201	38	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.	M

Section	SubSection	Page	No	Description	Status
25	25.1	221	1	Many SIP header field values consist of words separated by LWS or special characters. Unless otherwise stated, tokens are case-insensitive. These special characters **MUST** be in a quoted string to be used within a parameter value.	M
		223	2	The BNF for telephone-subscriber can be found in RFC 2806 [9]. Note, however, that any characters allowed there that are not allowed in the user part of the SIP URI **MUST** be escaped.	M

Section	SubSection	Page	No	Description	Status
26	26.2.1	239	1	The TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite [6]**MUST** be supported at a minimum by implementers when TLS is used in a SIP application.	M
26	26.2.1	239	2	For purposes of backwards compatibility, proxy servers, redirect servers, and registrars **SHOULD** support TLS_RSA_WITH_3DES_EDE_CBC_SHA. Implementers MAY also support any other ciphersuite.	S
26	26.2.2	240	3	The use of SIPS in particular entails that mutual TLS authentication**SHOULD** be employed, as **SHOULD** the ciphersuite TLS_RSA_WITH_AES_128_CBC_SHA.	S
			4		S
26	26.2.2	240	5	Certificates received in the authentication process**SHOULD** be validated with root certificates held by the client; failure to validate a certificate**SHOULD** result in the failure of the request.	S
			6		S
26	26.3.1	241	7	Proxy servers, redirect servers, and registrars**MUST** implement TLS, and **MUST** support both mutual and one-way authentication.	M
			8		M
26	26.3.1	241	9	It is strongly **RECOMMENDED** that UAs be capable initiating TLS; UAs MAY also be capable of acting as a TLS server.	R

Section	SubSection	Page	No	Description	Status
26	26.3.1	241	10	Proxy servers, redirect servers, and registrars**SHOULD** possess a site certificate whose subject corresponds to their canonical hostname.	S
26	26.3.1	241	11	All SIP elements that support TLS**MUST** have a mechanism for validating certificates received during TLS negotiation; this entails possession of one or more root certificates issued by certificate authorities (preferably well-known distributors of site certificates comparable to those that issue root certificates for web browsers).	M
26	26.3.1	241	12	All SIP elements that support TLS**MUST** also support the SIPS URI scheme.	M
26	26.3.1	241	13	When a UA attempts to contact a proxy server, redirect server, or registrar, the UAC **SHOULD** initiate a TLS connection over which it will send SIP messages. In some architectures, UASs MAY receive requests over such TLS connections as well.	S
26	26.3.1	241	14	Proxy servers, redirect servers, registrars, and UAs**MUST** implement Digest Authorization, encompassing all of the aspects required in 22.	M
26	26.3.1	241	15	Proxy servers, redirect servers, and registrars**SHOULD** be configured with at least one Digest realm, and at least one "realm" string supported by a given server **SHOULD** correspond to the server's hostname or domainname.	S
			16		S
26	26.3.1	241	17	If a UA holds one or more root certificates of certificate authorities in order to validate certificates for TLS or IPSec, it**SHOULD** be capable of reusing these to verify S/MIME certificates, as appropriate.	S
26	26.3.2.1	242	18	When a UA comes online and registers with its local administrative domain, it **SHOULD** establish a TLS connection with its registrar (Section 10 describes how the UA reaches its registrar).	S

Section	SubSection	Page	No	Description	Status
26	26.3.2.1	242	19	The registrar **SHOULD** offer a certificate to the UA, and the site identified by the certificate **MUST** correspond with the domain in which the UA intends to register; for example, if the UA intends to register the address-of-record 'alice@atlanta.com', the site certificate must identify a host within the atlanta.com domain (such as sip.atlanta.com).	M
			20		M
26	26.3.2.1	242	21	When it receives the TLS Certificate message, the UA**SHOULD** verify the certificate and inspect the site identified by the certificate.	S
26	26.3.2.1	242	22	If the certificate is invalid, revoked, or if it does not identify the appropriate party, the UA **MUST NOT** send the REGISTER message and otherwise proceed with the registration.	M
26	26.3.2.1	243	23	The UA then creates a REGISTER request that**SHOULD** be addressed to a Request-URI corresponding to the site certificate received from the registrar.	S
26	26.3.2.1	243	24	When the UA sends the REGISTER request over the existing TLS connection, the registrar **SHOULD** challenge the request with a 401 (Proxy Authentication Required) response. The "realm" parameter within the Proxy-Authenticate header field of the response **SHOULD** correspond to the domain previously given by the site certificate.	S
			25		S
26	26.3.2.1	243	26	When the UAC receives the challenge, it**SHOULD** either prompt the user for credentials or take an appropriate credential from a keyring corresponding to the "realm" parameter in the challenge.	S
26	26.3.2.1	243	27	The username of this credential**SHOULD** correspond with the "userinfo" portion of the URI in the To header field of the REGISTER request.	S

Section	SubSection	Page	No	Description	Status
26	26.3.2.1	243	28	Once the registration has been accepted by the registrar, the UA**SHOULD** leave this TLS connection open provided that the registrar also acts as the proxy server to which requests are sent for users in this administrative domain.	S
26	26.3.2.2	244	29	Assuming that the client has completed the registration process described in the preceding section, it **SHOULD** reuse the TLS connection to the local proxy server when it sends an INVITE request to another user.	S
26	26.3.2.2	244	30	The UA **SHOULD** reuse cached credentials in the INVITE to avoid prompting the user unnecessarily.	S
26	26.3.2.2	244	31	When the local outbound proxy server has validated the credentials presented by the UA in the INVITE, it**SHOULD** inspect the Request-URI to determine how the message should be routed (see [4]).	S
26	26.3.2.2	244	32	The local outbound proxy server at atlanta.com**SHOULD** therefore establish a TLS connection with the remote proxy server at biloxi.com.	S
26	26.3.2.2	244	33	Since both of the participants in this TLS connection are servers that possess site certificates, mutual TLS authentication**SHOULD** occur.	S
26	26.3.2.2	244	34	Each side of the connection **SHOULD** verify and inspect the certificate of the other, noting the domain name that appears in the certificate for comparison with the header fields of SIP messages.	S
26	26.3.2.2	244	35	The atlanta.com proxy server, for example,**SHOULD** verify at this stage that the certificate received from the remote side corresponds with the biloxi.com domain.	S
26	26.3.2.2	244	36	The proxy server at biloxi.com**SHOULD** inspect the certificate of the proxy server at atlanta.com in turn and compare the domain asserted by the certificate with the "domainname" portion of the From header field in the INVITE request.	S

Section	SubSection	Page	No	Description	Status
26	26.3.2.2	245	37	Once the INVITE has been approved by the biloxi proxy, the proxy server **SHOULD** identify the existing TLS channel, if any, associated with the user targeted by this request (in this case "bob@biloxi.com").	S
26	26.3.2.2	245	38	Before they forward the request, both proxy servers**SHOULD** add a Record-Route header field to the request so that all future requests in this dialog will pass through the proxy servers.	S
26	26.3.2.3	245	39	Alternatively, consider a UA asserting the identity "carol@chicago.com" that has no local outbound proxy. When Carol wishes to send an INVITE to "bob@biloxi.com", her UA **SHOULD** initiate a TLS connection with the biloxi proxy directly (using the mechanism described in [4] to determine how to best to reach the given Request-URI).	S
26	26.3.2.3	245	40	When her UA receives a certificate from the biloxi proxy, it**SHOULD** be verified normally before she passes her INVITE across the TLS connection.	S
26	26.3.2.3	246	41	Carol **SHOULD** then establish a TCP connection with the designated address and send a new INVITE with a Request-URI containing the received contact address (recomputing the signature in the body as the request is readied).	S
26	26.3.2.4	246	42	When the host on which a SIP proxy server is operating is routable from the public Internet, it **SHOULD** be deployed in an administrative domain with defensive operational policies (blocking source-routed traffic, preferably filtering ping traffic).	S
26	26.3.2.4	247	43	UAs and proxy servers **SHOULD** challenge questionable requests with only a single 401 (Unauthorized) or 407 (Proxy Authentication Required), forgoing the normal response retransmission algorithm, and thus behaving statelessly towards unauthenticated requests.	S
26	26.4.2	249	44	Another, more prosaic difficulty with the S/MIME mechanism is that it can result in very large messages, especially when the SIP tunneling mechanism described in Section 23.4 is used. For that reason, it is**RECOMMENDED** that TCP should be used as a transport protocol when S/MIME tunneling is employed.	R
26	26.4.4	250	45	To address these concerns, it is **RECOMMENDED** that recipients of a request whose Request-URI contains a SIP or SIPS URI inspect the To header field value to see if it contains a SIPS URI (though note that it does not constitute a breach of security if this URI has the same scheme but is not equivalent to the URI in the To header field).	R

Section	SubSection	Page	No	Description	Status
26	26.4.4	250	46	If the UAS has reason to believe that the scheme of the Request-URI has been improperly modified in transit, the UA**SHOULD** notify its user of a potential security breach. As a further measure to prevent downgrade attacks, entities that accept only SIPS requests MAY also refuse connections on insecure ports.	S
26	26.5	251	47	A user location service can infringe on the privacy of the recipient of a session invitation by divulging their specific whereabouts to the caller; an implementation consequently **SHOULD** be able to restrict, on a per-user basis, what kind of location and availability information is given out to certain classes of callers.	S

Section	SubSection	Page	No	Description	Status
27	27.1	252	1	o Name of the option tag. The name MAY be of any length, but**SHOULD** be no more than twenty characters long. The name**MUST** consist of alphanum (Section 25) characters only.	S
27	27.1	252	2		M

Section	SubSection	Page	No	Description	Status
28	28.1	259	1	o In RFC 2543, a proxy was not required to forward provisional responses from 101 to 199 upstream. This was changed to **MUST**.	M