8. General User Agent Behavior

8.1 UAC Behavior

8.1.1 Generating the Request

Để có thể tạo một bản tin request bởi UAC, điều tối thiểu phải có 6 trường thông tin: To, From, CSeq, Call-ID, Max-Forwards,and Via. Với 6 trường thông tin này sẽ cho ta một bản tin Request với cấu trúc đơn giản nhất. Bởi nó khi đi với nhau chúng sẽ dễ dàng định tuyến đưọc, thứ tự bản tin, địa chỉ, từ đó có thể xác thực.

8.1.1.1. Request-URI

Request URI là thông tin liên hệ của chặng tiếp theo trong lộ trình cuộc gọi

Tập hợp tuyến đường có sẵn là một tập hợp các URI được sắp xếp để xác định một chuỗi máy chủ mà UAC sẽ gửi các yêu cầu gửi đi bên ngoài hộp thoại. Thông thường, chúng được cấu hình trên UA bởi người dùng hoặc nhà cung cấp dịch vụ theo cách thủ công hoặc thông qua một số cơ chế không phải SIP khác. Khi một nhà cung cấp muốn định cấu hình một UA với một proxy gửi đi, thì KHUYẾN NGHỊ rằng điều này nên được thực hiện bằng cách cung cấp cho nó một tuyến đường có sẵn được thiết lập với một URI duy nhất, đó là của proxy gửi đi.

8.1.1.2 To

Truờng To trước hết chỉ định mong muốn

đối tuờng mà yêu cầu muốn truyền tải hoặc địa chỉ lưu hồ sơ của

người dùng hoặc tài nguyên cần sau khi thực hiện yêu cầu này. Điều này có thể hoặc có thể

không phải là người nhận cuối cùng của yêu cầu

+ Trường tiêu đề To

CÓ THỂ chứa URI SIP hoặc SIPS, nhưng nó cũng có thể sử dụng URI khác

lược đồ

* A UAC may learn how to populate the To header field for a particular

request in a number of way ( tự đặt, hoặc lấy trong một danh sách địa chỉ có sắn)

* the user will not enter a complete URI, but rather a string of digits

or letters

* sing the string to form the

user part of a SIP URI implies that the UA wishes the name to be

resolved in the domain to the right-hand side (RHS) of the at-sign in

the SIP URI => **Bảo mật**. and that the name is to be resolved in the

domain to the RHS of the at-sign

* Then, The RHS will frequently be the

home domain of the requestor, which allows for the home domain to

process the outgoing request

* This is useful for features like

"speed dial" that require interpretation of the user part in the home

domain

* The tel URL may be used when the UA does not wish to specify

the domain that should interpret a telephone number that has been

input by the user.

* 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. Since

no dialog is established, no tag is present.

8.1.1.3. From

* The From header field indicates the logical identity of the initiator

of the request, possibly the user's address-of-record. => Giống trường To ( bởi chứa URI, và tùy chọn hiển thị name)

* It is

used by SIP elements to determine which processing rules to apply to

a request (for example, automatic call rejection)

* From URI not contain IP addresses or the FQDN

of the host on which the UA is running, since these are not logical

names.

* Nên để chế đồ hiển thị tên là: “Anonymous”, với cú pháp đúng khi đó bản tin sẽ như thế này: <sip:thisis@anonymous.invalid>. Nếu như mong muốn ẩn danh client.
* Việc cung cấp có thể do Admin của local domain mà người dùng hoạt động hay thậm chí là do người dùng.
* If a particular UA is

used by multiple users, it might have switchable profiles that

include a URI corresponding to the identity of the profiled user.

* Yếu tố xác thực
* The From field MUST contain a new "tag" parameter, chosen by the UAC.
  + - 1. Call-ID
* The Call-ID header field acts as a unique identifier to group

together a series of messages. => Cần giống trong tất cả các bản in request và respone được gửi bởi UA => Giống trong mỗi khi Registration từ UA.

* 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

* All SIP UAs must have a means to guarantee that the Call-

ID header fields they produce will not be inadvertently generated by

any other UA.

* Trong một vài trường hợp khi yêu bị lỗi (VD cần xác thực) thì không đó không phải một yêu cầu mới, bởi thế Call-ID không thay đổi.\
* Use of cryptographically random identifiers ([RFC 1750](https://www.rfc-editor.org/rfc/rfc1750) [[12](https://www.rfc-editor.org/rfc/rfc3261#ref-12)]) in the

generation of Call-IDs is RECOMMENDED. => provides some

protection against session hijacking and reduces the likelihood of

unintentional Call-ID collisions.

form "localid@host"

Call-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6@foo.bar.com

* Call-IDs are case-sensitive and are simply compared byte-by-byte.
  + - 1. Cseq
* The CSeq header field serves as a way to identify and order

transactions. Bao gồm Sequence number và một phương thức. ( Điểm lưu ý ở đây là phương thức phải match với request)

* For non-REGISTER requests

outside of a dialog, the sequence number value is arbitrary

* he
* The sequence number value MUST be expressible as a 32-bit unsigned

integer and MUST be less than 231

Ex: CSeq: 4711 INVITE

* + - 1. Max-Forwards
* The Max-Forwards header field serves to limit the number of hops a request can transit on the way to its destination. Với cơ chế tự động mỗi khi qua một hop nó sẽ giảm một số nguyên lần là 1.
* If the Max-Forward value reaches 0 before the request reaches its destination, it will be rejected with a 483(Too Many Hops) error response.
* A UAC MUST insert a Max-Forwards header field into each request it originates with a value that SHOULD be 70 => Đủ lớn để không để trường hợp bị drop trên mạng khi mà không xảy ra hiện tượng loops. Nhưng nếu quá lớn thì lại tiêu tốn tài nguyên proxy khi loop xảy ra.
* Việc tối ưu phải do UA nắm rõ nhiều yếu tố, trong đó là cấu hình mạng (topologies)
  + - 1. Via
* The Via header field indicates the transport used for the transaction

and identifies the location where the response is to be sent. A Via

header field value is added only after the transport that will be

used to reach the next hop has been selected

* When the UAC creates a request, it MUST insert a Via into that

request. The protocol name and protocol version in the header field

MUST be SIP and 2.0, respectively. The Via header field value MUST

contain a branch parameter

* This parameter is used to identify the transaction created by that request. This parameter is used by both the client and the server.
* The branch parameter value MUST be unique across space and time for all requests sent by the UA. (ngoại trừ bản tin CANCEL và non-2xx respone).
* The uniqueness property of the branch ID parameter, to facilitate

its use as a transaction ID.

* The branch ID inserted by an element compliant with this specification MUST always begin with the characters "z9hG4bK"
* These 7 characters are used as a magic cookie (7 is deemed sufficient to ensure that an older implementation would not pick such a value), so that servers receiving the request can determine that the branch ID was constructed in the fashion described by this specification (that is, globally unique)
  + - 1. Contact
* The Contact header field provides a SIP or SIPS URI that can be used

to contact that specific instance of the UA for subsequent requests.

* The Contact header field MUST be present and contain exactly one SIP

or SIPS URI in any request

* For the methods defined in this specification, that includes

only the INVITE request. For these requests, the scope of the

Contact is global.

* 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.

* Trong trường hợp Request-URI hoặc trường Route header chứa SIPS URI, trường Contact cũng phải chứa giống như vậy.
  + - 1. Supported and Require
* Nếu muốn cung cấp thêm hỗ trợ bởi UAC, liệt kê thêm trong trường tags, tương ứng với từng thành phần hỗ trợ thêm đó. => Để server không thực hiện nhưng dịch vụ không cần thiết từ đó tránh lãng phí hiệu suất làm việc.

Nếu như UAC mong muốn phía UAS hiểu rõ ràng hơn về yêu cầu nó muốn truyền tải, thì nó sẽ cung cấp thêm trường thông tin Require(into the request listing the option tag for that extension)

* Trong trường hợp dành riềng cho Proxies, phải chèn trường thông tin Proxy-Require (into the request listing the option tag for that extension.
* Lưu ý các quy tắc option tags trong Require va Proxy-Require áp dụng tương tựu như Supported.
  + - 1. Additional Message Components
* Xem thêm tại <https://www.rfc-editor.org/rfc/rfc3261#section-20.11>
  + 1. Sending the Request
* Xét trong trường hợp không cần thêm “local policy specifying” nào khác. Khí đó đích sẽ được xác định bởi thủ tục DNS (<https://www.rfc-editor.org/rfc/rfc3261#ref-4> ).
* 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](https://www.rfc-editor.org/rfc/rfc3261#section-12.2.1.1)), the procedures MUST be applied to the Request-URI of the request. therwise, the procedures are applied to the first Route header field mvalue in the request (if one exists), or to the request's Request-URI mif there is no Route header field present. (Đọc lại sau!)
* Thứ tự xét khi gửi một bản tin request là: address, port, and transports

*These procedures yield an ordered set of address, port, and transports to attempt. Independent of which URI is used as input to the procedures of [*[*4*](https://www.rfc-editor.org/rfc/rfc3261#ref-4)*], if the MRequest-URI specifies a SIPS resource, the UAC MUST follow the Mprocedures of [*[*4*](https://www.rfc-editor.org/rfc/rfc3261#ref-4)*] as if the input URI were a SIPS URI ???*

* 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. Beyond that, there are no restrictions on the alternate destinations if the request contains no Route header field
* Có 2 cách thay thế: outbound proxy và thay thế single URI. Cách thứ 2 nên dùng hơn( dưa trên pre-existing)
* 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
* **Quy trình**
* 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.

* + 1. Processing Response
* Là tiến trình đầu tiên tại lớp transport layer và sau đó chuyển lên lớp cao hơn là transaction layer. Tới lớp transaction, sẽ thực hiện quy trình của xử lý sau khi nhận từ lớp dưới và chuyển phản hổi ngược trở lại TU.

8.1.3.1 Transaction Layer Errors

* Trong một số trường hợp, bản tin response trả lại không phải bản tin SIP mà là một TLE:

+ Khi timeout error được nhận từ TL, nó phải hành động như xảy ra lỗi 408(Request Timeout)

+ Khi fatal transport error, nó hành động như xảy ra lỗi 503(Service Unavailable)

8.1.3.2 Unrecognized Responses

* UAC khi nhận được một bản tin final responses không rõ ràng, không hiểu được – x00 response code. Nó có cách xử lý cho tất cả trường hợp này.

VD: Nếu nó nhận được bản tin 431, nó không hiểu được, và để an toàn nó cho rằng đã nhận 400 (Bad Request) thay vì 431.

8.1.3.3 Vias

* Trong trường hợp nhiều hơn một trường Via trong phản hổi, UAC sẽ ngay lập trưc hủy đi.
* Việc định tuyến sai, hoặc bản tin có thể bị hỏng.
  + - 1. Processing 3xx Responses
* 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. A client starts with an initial target set containing exactly one URI, the Request-URI of the original request. If a client wishes to formulate new requests based on a 3xx class response to that request, it placesthe URIs to try into the target set. Subject to the restrictions in this specification, a client can choose which Contact URIs it placesinto the target set

* A common mechanism is to order the set by the "q" parameter value from the Contact header field value
* If contacting an address in the list results in a failure, as defined

in the next paragraph, the element moves to the next address in the

list, until the list is exhausted. If the list is exhausted, then

the request has failed.

* 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

* When a failure for a particular contact address is received, the client SHOULD try the next contact address – tạo một new client transaction để truyền đạt một request mới.
* **Cách tạo request**
* a UAC MUST copy the entire URI from the target set into the

Request-URI, except for the "method-param" and "header" URI

Parameters

* It uses the "header" parameters to create header field values for the

new request, overwriting header field values associated with the

redirected request.

* Trong một số trường hợp trường header có thể không cần viết đè, mà có thể viết nối tiếp và ngăn cách bằng dấu phẩy. (Với điều kiện hệ thống có quy ước chấp nhận), còn không sẽ như bước ở trên.



* But the HTTP URL is merely appended to any existing Call-Info header field values.
* Trong thực tế, UAC sẽ tái sử dụng lại các trường thông tin lúc trước To, From and Call-ID có trong original redirected request. Nhưng hoàn toàn có thể update Call-ID field. Cũng tái sử dụng SDP.
* Sau khi update, nó chắc chắn nhận được một ID mới.
  + - 1. Processing 4xx Responses
* If a 401 (Unauthorized) or 407 (Proxy Authentication Required) response is received => <https://www.rfc-editor.org/rfc/rfc3261#section-22.2>

( Chưa đọc )

* If a 413 (Request Entity Too Large) response is received, the request contained a body that was longer than the UA was willing to accept
* Truyền lại request, hoặc bỏ phần thân, thậm chí có thể giảm chiều dài yêu cầu.
* If a 415 (Unsupported Media Type) response is received, the request contained media types not supported by the UAS.
* Gửi lại request với mã hóa, nội dung phù hợp với cấu trúc
* If a 416 (Unsupported URI Scheme) response is received , the Request-URI used a URI scheme not supported by the server.
* Client nên gửi thử lại request, nhưng sử dụng SIP URI
* If a 420 (Bad Extension) response is received the request contained a Require or Proxy-Require header field listing an option-tag for a feature not supported by a proxy or UAS.
* UAC nên gửi lại request, lược bỏ bỏ các phần mở rộng không được hỗ trợ trong phản hồi.
* Có thầy tất cả các tình huống ở trên, đều tạo một request mới, nhưng điều ta lưu ý ở đây là các trường cơ bản vẫn giữ nguyên như Call-ID, To, From. Nhưng Cseq lên tăng lên 1 so với giá trị trước đó.
  1. UAS Behavior
     1. Method Inspection
* Ngay sau khi một request được chứng thực (hoặc bị skipped), UAS phải kiểm tra method của request. Nếu UAS nhận thấy rằng không thể hỗ trwoj method cho request, nó lập tức tạo 405 (Method Not Allowed) reponse.
* The UAS MUST also add an Allow header field to the 405 (Method Not Allowed) response. The Allow header field MUST list the set of methods supported by the UAS generating the message
* Còn nếu được hỗ trợ bở sever, tiến trình vẫn sẽ tiếp tục.
  + 1. Header Inspection
* If a UAS does not understand a header field in a request (có thể do chưa được định nghĩa cụ thể).
* the server MUST ignore that header field and

continue processing the message. A UAS SHOULD ignore any malformed

header fields that are not necessary for processing requests.

8.2.2.1 To and Request-URI

* The To header field identifies the original recipient of the request

designated by the user identified in the From field

* Therefore, The original recipient may or may not be the UAS processing the request, due to call forwarding or other proxy operations. 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
* Trong thực tế nó có thể nhận kể cả khi nó không biết rõ, nhưng nếu từ chối nó sẽ gửi bản tin 403 (Forbidden)
* Còn đối với Request-URI, nếu trường hợp không được hỗ trợ bởi UAS, từ chối và trả lại bản tin 416 (Unsuppoerted URI Scheme) response.
* Nếu Request-URI không định dạng rõ địa chỉ UAS chấp nhận, UAS sẽ trả lại bản tin 404 (Not Found) response.

8.2.2.2 Merged requests

* If the request has no tag in the To header field, the UAS core MUST

check the request against ongoing transactions. If the From tag,

Call-ID, and CSeq exactly match those associated with an ongoing

transaction, but the request does not match that transaction (based

on the matching rules in [Section 17.2.3](https://www.rfc-editor.org/rfc/rfc3261#section-17.2.3)), the UAS core SHOULD

generate a 482 (Loop Detected) response and pass it to the server

transaction.

8.2.2.3 Require

* The Require header field is used by a UAC to tell a UAS about SIP

extensions that the UAC expects the UAS to support in order to

process the request properly. Its format is described in [Section](https://www.rfc-editor.org/rfc/rfc3261#section-20.32)

[20.32](https://www.rfc-editor.org/rfc/rfc3261#section-20.32). If a UAS does not understand an option-tag listed in a

Require header field, it MUST respond by generating a response with

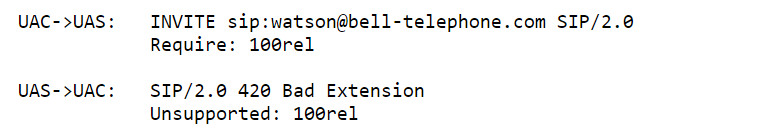
status code 420 (Bad Extension). The UAS MUST add an Unsupported

header field, and list in it those options it does not understand

amongst those in the Require header field of the request

* Lưu ý rằng Require và Proxy-Require không được sử dụng trong SIP CANCEL request, hoặc trong một ACK request đã được gửi với non-2xx response.
* These header fields MUST be ignored if they are present in these requests.
* An ACK request for a 2xx response MUST contain only those Require and

Proxy-Require values that were present in the initial request.

****

* Giảm delay, đồng thời phản ứng với các trường hợp extension không rõ rằng
  + 1. Content Processing
* Assuming the UAS understands any extensions required by the client,

the UAS examines the body of the message, and the header fields that

describe it. If there are any bodies whose type (indicated by the

Content-Type), language (indicated by the Content-Language) or

encoding (indicated by the Content-Encoding) are not understood, and

that body part is not optional (as indicated by the Content-

Disposition header field)

* Phản hồi bản tin 415 (Unsupported Media Type) response
* The response MUST contain an Accept header field listing the types of all bodies it understands, in the event the request contained bodies of types not supported by the UAS.
  + If the request contained content encodings not understood by the UAS, the response MUST contain an Accept-Encoding header field listing the encodings understood by the UAS
  + If the request contained content with languages not understood by the UAS, the response MUST contain an Accept-Language header field indicating the languages understood by the UAS.
    1. Applying Extensions
* 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. If the

desired extension is not supported, the server SHOULD rely only on

baseline SIP and any other extensions supported by the client.

* Gửi bản tin 421 (Extension Required)
* Bản tin ở trên ngầm hiều là phía Sever không thể thực hiện được nếu không đủ thêm thông tin.
* Thực tế, là sever chấp nhận cung cấp không trọn vẹn.
  + 1. Processing the Request
* Assuming all of the checks in the previous subsections are passed,

the UAS processing becomes method-specific. [Section 10](https://www.rfc-editor.org/rfc/rfc3261#section-10) covers the

REGISTER request, [Section 11](https://www.rfc-editor.org/rfc/rfc3261#section-11) covers the OPTIONS request, [Section 13](https://www.rfc-editor.org/rfc/rfc3261#section-13)

covers the INVITE request, and [Section 15](https://www.rfc-editor.org/rfc/rfc3261#section-15) covers the BYE request.

* + 1. Gerating the Response

8.2.6.1 Sending a Provisional Response

* 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

present in the request MUST be copied into this 100 (Trying)

response

* If there is a delay in generating the response, the UAS

SHOULD add a delay value into the Timestamp value in the response

* Các giá trị time cho response hay receipt khác nhau và được đo lường bằng seconds.

8.2.6.2 Header and Tags

* The From field of the response MUST equal the From header field of

the request. The Call-ID header field of the response MUST equal the

Call-ID header field of the request. The CSeq header field of the

response MUST equal the CSeq field of the request. The Via header

field values in the response MUST equal the Via header field values

in the request and MUST maintain the same ordering

* If a request contained a To tag in the request, the To header field

in the response MUST equal that of the request

* Trong trường hợp không có có tag, UAS phải cho URI bằng với request, ngoài ra phải thêm trường tag vào trường To trong phản hồi (Ngoại trừ với 100 Trying)
  + 1. Stateless UAS Behavior
* A stateless UAS is a UAS that does not maintain transaction state.
* It replies to requests normally, but discards any state that would

ordinarily be retained by a UAS after a response has been sent.

* If a stateless UAS receives a retransmission of a request, it regenerates

the response and resends it, just as if it were replying to the first

instance of the request

* Sử dụng cho các trường hợp yêu cầu không cần chứng thực, bởi nếu trường hợp này sử dụng cho statefull nó sẽ gây ra overflow.\
* **Nhưng điểm quan trọng:**

A stateless UAS MUST NOT send provisional (1xx) responses.

o A stateless UAS MUST NOT retransmit responses.

o A stateless UAS MUST ignore ACK requests.

o A stateless UAS MUST ignore CANCEL requests.

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](https://www.rfc-editor.org/rfc/rfc3261#section-19.3).

In all other respects, a stateless UAS behaves in the same manner as

a stateful UAS. A UAS can operate in either a stateful or stateless

mode for each new request.

* 1. Redirect Servers
* Mục đích để giảm lưu lượng tải trong các proxy, đồng thời cải thiện tín hiệu robustness.
* Sever -> gửi a new routing information (database) -> client -> tìm URIs mới -> Redirect.
* Xem thêm chi tiết tại <https://www.rfc-editor.org/rfc/rfc3261#section-8.3>

9. Canceling a Request

* Dùng để hủy yêu cầu do client gửi trước đó.
* Nó yêu cầu UAS ngừng xử lý yêu cầu, và tạo ra một error response -> không ảnh hưởng tới các final response của UAS. => Sử dụng kết thúc INVITE mà mất nhiều thời gian setup.
* In that usage, a UAS that receives a CANCEL request for an INVITE, but has not yet sent a final response, would "stop ringing", and then respond to the INVITE with a specific error response (a 487).
* CANCEL requests can be constructed and sent by both proxies and user

agent clients

* CANCEL được sử dụng trong “hop-by-hop”

9.1 Client Behavior

* Bản tin Cancel không nên gửi cho trường hợp khác ngoài dành cho INVITE.
* Vì luôn phản hồi ngay lập túc, từ đó xảy ra xung đột bởi nhiều điểm.\
* Matching các trường thông tin, và đặc biệt Via phải phù hợp với Request để bản tin Cancel hủy đúng yêu cầu mong muốn.
* If the request being cancelled contains a Route header field, the

CANCEL request MUST include that Route header field's values.

* This is needed so that stateless proxies are able to route CANCEL

requests properly

* Để gửi -> chờ quyết định, vì xung đột -> Ảnh hưởng tới UAS
* Note that both the transaction corresponding to the original request and the CANCEL transaction will complete independently. However, a UAC canceling a request cannot rely on receiving a 487 (Request Terminated) response for the original request, as an [RFC 2543](https://www.rfc-editor.org/rfc/rfc2543)- compliant UAS will not generate such a response. If there is no final response for the original request in 64\*T1 seconds (T1 is defined in [Section 17.1.1.1](https://www.rfc-editor.org/rfc/rfc3261#section-17.1.1.1)), the client SHOULD then consider the original transaction cancelled and SHOULD destroy the clienttransaction handling the original request.

9.2 Server Behavior

* Quy trình CANCEL có thể khác nhau đối với từng loại Sever khác nhau:
* The processing of a CANCEL request at a server depends on the type of

server. A stateless proxy will forward it, a stateful proxy might

respond to it and generate some CANCEL requests of its own, and a UAS

will respond to it.

* A UAS first processes the CANCEL request according to the general UAS

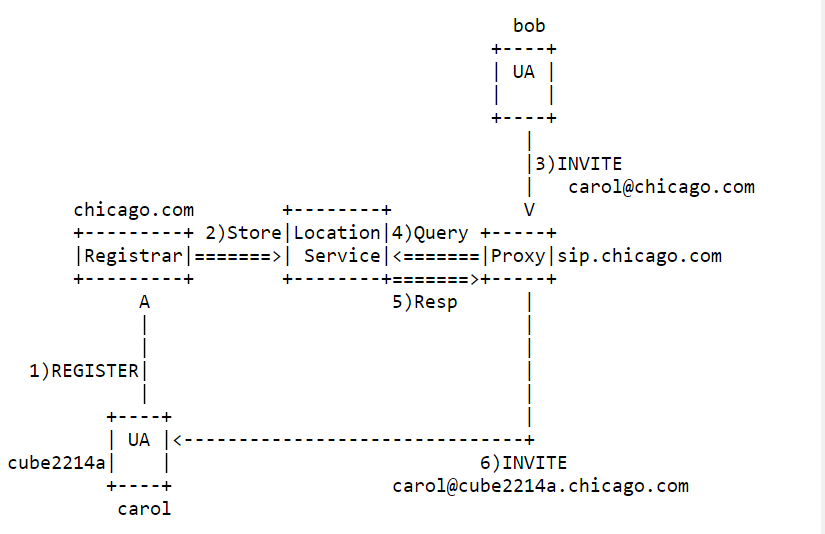
processing described in [Section 8.2](https://www.rfc-editor.org/rfc/rfc3261#section-8.2).

* Tuy nhiên điều đó không được thực hiện, bởi ta đã nói lúc trước chỉ áp dụng hop-by-hop. Và CANCEL requests không chứa Require header.
* Trong trường hợp không tìm thấy thông tin matching => gửi bản tin 481 (Call Leg/ Transaction Does not Exist)
* If the transaction for the original request still exists, the behavior of the UAS on receiving a CANCEL request depends on whether it has already sent a final response for the original request.
* Nếu có, thì không ảnh hưởng đến việc xử lý và trạng thái phiên, hay các phản hồi trước đó
* Nếu chưa, phụ thuộc vào method của original request. Nếu là INVITE sẽ trả ngay lập tức 487
* Điều quan tỏng phải matching thông tin để có thể hoàn toàn hủy. (To tag = nhau )

10. Registrations

10.1 Overview

=> OK



10.2 Constructing the REGISTER request

* REGISTER requests add, remove, and query bindings. A REGISTER

request can add a new binding between an address-of-record and one or

more contact addresses. Registration on behalf of a particular

address-of-record can be performed by a suitably authorized third

party. A client can also remove previous bindings or query to

determine which bindings are currently in place for an address-of-

record.

* Không thành thập một a dialog.
* 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

* Các trường thông tin phải có:

|  |  |
| --- | --- |
| Request-URI: | The Request-URI names the domain of the location service for which the registration is meant  The "userinfo" and "@" components of the SIP URI MUST NOT be present. |
| To | The To header field contains the address of record whose  registration is to be created, queried, or modified. |
| 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. |
| Call-ID: | All registrations from a UAC SHOULD use the same Call-ID header field value for registrations sent to a particular registrar.   * Nếu không cùng thì việc tìm hay trì hoãn đăng kí sẽ không có sự công bằng. |
| CSeq | The CSeq value guarantees proper ordering of REGISTER requests.   * Qua mỗi hop sẽ tăng lên một |
| Contact | EGISTER requests MAY contain a Contact header field with zero or more values containing address bindings. |

* Điểm lưu ý ở đay là UA không được gửi bản tin đăng kí mói khi mà bản tin final respond chưa được nhận do registrar gửi hoặc chưa hết time-out bản tin đăng kí trước đó.