Abstract

This document defines two SIP extension header fields and associated option tags that can be used in INVITE requests to convey the requester’s preference for user-interface handling of that request. The first header, "Answer-Mode", expresses a preference as to whether the target node’s user interface waits for user input before accepting the request or instead accepts the request without waiting.
on user input. The second header, "Alert-Mode", expresses a preference as to whether the target node’s user interface alerts the user about the request. These behaviors have applicability to applications such as Push-to-Talk and to diagnostics like loop-back. This document also defines use of the SIP extension header field "Answer-Mode", in a response to an INVITE request to inform the requester as to which answer mode was actually applied to this request. There are significant security considerations, especially when the two request options are used together.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].
Table of Contents

1. Background .................................................. 5

2. Requirements .................................................. 6
   2.1 Requirements for Requesting an Answering Mode .......... 6
   2.2 Requirements for Requesting an Alerting Mode ......... 8
   2.3 Requirements for Indicating the Applied Answer Mode in a Response ......................... 8

3. Syntax of Header Fields and Tags ............................. 9
   3.1 Syntax of Header Field and Tags ......................... 9
   3.2 Amendments to Table 2 and 3 of RFC3261 ................. 9

4. Usage of the Answer-Mode Header Field, Option, and Media Feature Tags in a Request ...................... 10
   4.1 Procedures at the UAC ................................. 10
      4.1.1 All Requests ........................................ 10
      4.1.2 REGISTER Transactions .............................. 10
      4.1.3 INVITE Transactions ................................. 11
   4.2 Procedures at Intermediate Proxies .................... 12
   4.3 Procedures at the UAS .................................. 12
   4.4 Issues with Automatic Answering and Forking ........... 13

5. Usage of the Alert-Mode Header Field, Option, and Media Feature Tags in a Request ...................... 14
   5.1 Procedures at the UAC .................................. 14
      5.1.1 All Requests ........................................ 14
      5.1.2 REGISTER Transactions .............................. 14
      5.1.3 INVITE transactions ................................. 14
   5.2 Procedures at Intermediate Proxies .................... 15
   5.3 Procedures at the UAS .................................. 15

6. Usage of the Answer-Mode Header Field in a Response ...... 15
   6.1 Procedures at the UAS .................................. 16
   6.2 Procedures at the UAC .................................. 16

7. Examples of Usage ............................................. 16
   7.1 REGISTER Request ....................................... 17
   7.2 INVITE Request .......................................... 17
   7.3 200 OK response ........................................ 17

8. Security Considerations ...................................... 17

9. IANA Considerations ........................................ 18
   9.1 Registration of Header Fields ............................ 18
   9.2 Registration of Header Field Parameters ................ 19
   9.3 Registration of Extension Option Tags ................... 19
10.   Acknowledgements  ...........................................  20

11.   References  .................................................  20
11.1  Normative References  ......................................  20
11.2  Informative References  .................................  21

Authors’ Addresses  ..............................................  22

Intellectual Property and Copyright Statements  ............  23
1. Background

There has been discussion of how to deal with "auto-answer" and related issues in the SIP community for several years. Discussion in the SIPPING working group, augmented by input from other organizations such as the Open Mobile Alliance, resulted in a consensus observed in the SIPPING meeting at IETF 62 to extend SIP, which is defined in [2]. Further discussion of the topic on the SIP mailing list after IETF 62 led to a consensus to pursue this work in the SIP working group as a standards-track effort.

Two different use cases converged to create the consensus for the development of this specification. Other use cases presumably exist, but two is enough to establish the level of reusability required to justify a standards-track extension as opposed to a "P-header" under [3].

The first key use case was the requirement for diagnostic loopback calls. In this sort of scenario, a testing service sends an INVITE to a node being tested. The tested node accepts and a dialog is established. But rather than establishing a two-way media flow, the tested node loops back or "echoes" media received from the testing service back toward the testing service. The testing service can then analyze the media flow for quality and timing characteristics. SDP usage for this sort of flow is described in [11]. In this sort of application, it may not be needful that the human using the node under test interact with the node in any way for the test to be satisfactorily executed. In some cases, it might be appropriate to alert the user to the ongoing test, and in other cases it might not be.

The second use case is that of "Push to Talk" applications as described in [12] and relates to a service being specified by the Open Mobile Alliance. In this sort of environment, SIP is used to establish a dialog supporting asynchronous delivery of unidirectional media flow, giving a user experience like that of a traditional two-way radio. It is conventional for the INVITES used to be automatically accepted by the called UA (User Agent), and the media is commonly played out on a loudspeaker.

These sorts of mechanisms are not required to provide the functionality of an "answering machine" or "voice mail recorder". Such a device knows that it should answer and does not require a SIP extension to support its behavior.

Much of the discussion of this topic in working group meetings and on the mailing list dealt with disambiguating "answering mode" from "alerting mode". Some early work, such as [12], did not make this
distinction. We therefore proceed with the following definitions:

- Answering Mode includes behaviors in a SIP UA relating to acceptance or rejection of a request that are contingent on interaction between the UA and the user of that UA after the UA has received the request. We are principally concerned with the user interaction involved in accepting the request and initiating an active session. An example of this might be pressing the "yes" button on a mobile phone.

- Alerting Mode includes behaviors in a SIP UA relating to informing the user of the UA that a request to initiate a session has been received. An example of this might be activating the ring tone of a mobile phone.

2. Requirements

Requirements in the following are expressed relative to the node initiating an INVITE request (UAC), the node receiving and potentially responding to that request (UAS), and the users of those nodes (UAC-user and UAS-user).

2.1 Requirements for Requesting an Answering Mode

The requirements relating to requesting a specific answering mode include:

Req-1: It MUST be possible for UAC to ask that the UAS answer the request without requiring interaction between UAS-user and the user interface (UI) of the UAS. We refer to this as "automatic answer mode". This mode is useful for diagnostic loopback procedures and critical for "two-way radio" or "push to talk" applications.

Req-2: It MUST be possible for UAC to ask that the UAS answer the request only after UAS-user has directed UAS to answer this specific request. We refer to this as "manual answer mode". This mode is useful in "push to talk" applications where the sender requires a reassurance that somebody is listening.

Req-3: It MUST be possible for UAS to apply local policy to each request and determine whether or not to provide the requested answer mode for this request. This policy determination MAY include authentication checks, authorization against "buddy lists" as used in some presence systems, or other mechanisms outside the scope of this specification. This behavior is critical in avoiding major security pitfalls, such as turning the victim’s phone into a "bug" or eavesdropping device.
Req-4: It MUST be possible for UAC to indicate in the request that this extension for selecting answering mode is required, such that UAS MUST reject the request if it does not support this extension. This can be used to prevent automated diagnostic loopback requests from annoying nodes not supporting this extension.

Req-5: It MUST be possible for UAC to indicate at least two different priority levels for the desired answer mode. We refer to these as "normal" and "override" priorities. In normal usage, we expect that "normal" priority would be used in a user-to-user fashion, whereas "override" priorities would be used for diagnostic procedures or some sorts of emergency session establishment. This behavior allows a device to be set up such that it might not auto-answer routine calls, but could be convinced to auto-answer an emergency or other high-priority call.

Req-6: It MUST be possible for UAS or proxies acting on behalf of UAS to apply policy relative to the indicated priority level. This MAY include having different authentication and or authorization procedures for each priority level. This capability allows functions like time-of-day call screening, so that routine calls that would normally be rejected locally by the device would be blocked by a proxy without access network costs, but high-priority calls that would override routine call screening could be passed to the device.

Req-7: It MUST be possible for UAS to indicate its support for the selection of answer modes in a REGISTER request so that the routing proxy can selectively route requests requiring the selection of answer mode to UAS. This requirement enables the functions described in the next requirement.

Req-8: It MUST be possible for the UAC to construct the request in such a way that the routing proxy infrastructure, if present, will select only contacts supporting the selection of answer modes. This can efficiently (minimal access network traffic and minimal forking load) prevent devices that do not support this extension from being reached by requests that require this extension. Note that this requirement does NOT include selection of a singular UAS from a set to which the request might be forked.

Req-9: It MUST be possible for UAC to discover whether UAS supports the selection of answer modes via a SIP OPTIONS request.

Req-10: It MUST be possible for an intermediate proxy acting on behalf of UAC or UAS to apply policy relative to the answer mode indicated in a request. For example, a proxy may require special authentication and authorization for a request that places a high priority on auto-answer capabilities. Application of policy here means altering the requested answer mode and/or inserting or deleting a request for a specific answer mode.
2.2 Requirements for Requesting an Alerting Mode

The requirements relating to requesting a specific alerting mode include:

Req-11: It MUST be possible for UAC to ask that UAS answer the request without alerting UAS-user. This allows for diagnostic loopbacks that do not needlessly interrupt the user of a device.

Req-12: It MUST be possible for UAS to apply local policy to each request and determine whether or not to provide the requested alerting mode for this request. This policy determination MAY include authentication checks, authorization against "buddy lists" as used in some presence systems, or other mechanisms outside the scope of this specification.

Req-13: It MUST be possible for UAC to indicate in the request that this extension for selecting alerting mode is required, such that UAS MUST reject the request if it does not support this extension. This capability augments the ability of automated testing functions to operate non-intrusively when some devices in a network do not support this extension.

Req-14: It MUST be possible for UAC to discover whether UAS supports the selection of alerting modes via a SIP OPTIONS request.

Req-15: It MUST be possible for UAS to indicate its support for the selection of alerting modes in a REGISTER request so that the routing proxy can selectively route requests requiring the selection of alerting mode to UAS. This supports the functionality described in the following requirement.

Req-16: It MUST be possible for UAC to construct the request in such a way that the routing proxy infrastructure, if present, will select only contacts supporting the selection of alerting modes. This allows the proxy network to efficiently avoid sending the request to nodes that do not support this extension.

Req-17: It MUST be possible for an intermediate proxy acting on behalf of UAC or UAS to apply policy relative to the alerting mode indicated in a request. Application of policy here means altering the requested alerting mode and/or inserting or deleting a request for a specific alerting mode.

2.3 Requirements for Indicating the Applied Answer Mode in a Response

The requirements relating to indicating which answering mode applied to the request include:

Req-18: It MUST be possible for UAS when sending a positive response to a request to indicate the answering mode that applied to the request. This allows UAC to inform UAC-user as to whether the request was answered automatically or as a result of user interaction, knowledge that may be important in informing UAC-user.
user’s usage of the session.

Req-19: UAS SHOULD accurately represent the answering mode that was
applied, but MAY either not include this information or MAY
misinform UAC in order to maintain the privacy expectations of
UAS-user. Consequently, applications MUST NOT rely on the
veracity of this information.

3. Syntax of Header Fields and Tags

3.1 Syntax of Header Field and Tags

The syntax for the header fields defined in this document is:

Answer-Mode = "Answer-Mode" HCOLON answer-mode

Alert-Mode = "Alert-Mode" HCOLON alert-mode
alert-mode = "Normal" / "Null"

The syntax of the Alert-Mode option tag is:

Alert-Mode = "alertmode"

The syntax of the Answer-Mode option tag is:

Answer-Mode = "answermode"

The syntax of the feature tag indicating support for selection of the
answer mode is:

Answer-Mode = "answermode"

The value range of the Answer-Mode feature tag is binary, with
values of "TRUE" or "FALSE".

3.2 Amendments to Table 2 and 3 of RFC3261

The allowable usage of header fields is described in Tables 2 and 3
of [2]. The following additions to this table are needed for the
extension header fields defined in this document.

Additions to SIP Table 3:

<table>
<thead>
<tr>
<th>Header field</th>
<th>where</th>
<th>proxy</th>
<th>ACK</th>
<th>BYE</th>
<th>CAN</th>
<th>INV</th>
<th>OPT</th>
<th>REG</th>
<th>PRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer-Mode</td>
<td>I</td>
<td>adm</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Alert-Mode</td>
<td>I</td>
<td>adm</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Answer-Mode</td>
<td>200</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
4. Usage of the Answer-Mode Header Field, Option, and Media Feature Tags in a Request

The Answer-Mode header field is used by a UAC to request specific handling of an INVITE request by the responding UAS related to "automatic answering" functionality. If no Answer-Mode header field is included in the request, answering behavior is at the discretion of the UAS, as it would be in the absence of this specification. The desired handling is indicated by the the value of the Answer-Mode header field, as follows:

Manual: The UAS is asked to not accept the request (send a 200 OK) until the user of the UAS has interacted with the user interface (UI) of the UAS in such a way as to indicate that the user desires the UAS to accept the request.

ManualReq: The UAS is strongly asked to accept the request manually, as in "Manual". Further, the UAS is asked to override local user preferences relating to automatic answer, and answer manually even if the user preferences are to automatically answer requests having a Answer-Mode header field value of "Manual". The UAS is also asked NOT to answer automatically, and to reject the request if it is unwilling to answer manually.

Auto: The UAS is asked to accept the request automatically, without waiting for the user of the UAS to interact with the UI of the UAS in such a way as to indicate that the user desires the UAS to accept the request.

AutoReq: The UAS is strongly asked to accept the request automatically, as in "Auto". Further, the UAS is asked to override local user preferences relating to automatic answer, and answer automatically even if the user preferences are to not automatically answer requests having a Answer-Mode header field value of "Auto". The UAS is also asked NOT to answer manually, and to reject the request if it is unwilling to answer automatically.

4.1 Procedures at the UAC

4.1.1 All Requests

A UAC supporting this specification indicates its support for this extension by including an option tag of "answermode" in the Supported header field of all requests it sends.

4.1.2 REGISTER Transactions

To indicate that it supports the answer-mode negotiation feature, a
UA includes a SIP extension feature tag of "answermode" in the Contact: header field of its REGISTER requests. This usage of feature tags is described in [5].

4.1.3 INVITE Transactions

A UAC supporting this specification includes a Answer-Mode header field and appropriate value in an INVITE where it wishes to influence the answering mode of the responding UAS.

To request that the UAS answer only after having interacted with its user and receiving an affirmative instruction from that user, the UAC includes a Answer-Mode header field having a value of "Manual".

To request that the UAS answer manually, and ask that it reject the INVITE request if unable or unwilling to answer manually, the UAC includes a Answer-Mode header field having a value of "ManualReq".

To request that the UAS answer automatically without waiting for input from the user, the UAC includes a Answer-Mode header field having a value of "Auto".

To request that the UAS answer automatically, and ask that it reject the INVITE request if unable or unwilling to answer automatically, the UAC includes a Answer-Mode header field having a value of "AutoReq".

To require that the UAS either support this extension or reject the request, the UAC includes a Required: header field having the value "answermode". Note that this does not actually force the UAS to automatically answer, it just requires that the UAS understand this negotiation mechanism. We do not have a negotiation technique (like "requires") to force specific behavior. Rather, the desired behavior is indicated in the SIP extension itself.

To request that retargeting proxies in the path preferentially select targets that have indicated support for this extension in their registration, a UAC includes an Accept-Contact header field having a parameter of "answermode". This usage of Accept-Contact is described in [6].

To request that retargeting proxies in the path do not select targets that have indicated non-support for this extension in their registration, a UAC includes an Accept-Contact header field having a parameter of "answermode" and an option field of "require". This usage of Accept-Contact is described in [6].

To request that retargeting proxies in the path exclusively select
targets that have indicated support for this extension in their registration, a UAC includes an Accept-Contact header field having a parameter of "answermode" and option fields of "require" and "explicit". This usage of Accept-Contact is described in [6].

4.2 Procedures at Intermediate Proxies

The general procedure at all intermediate proxies including the UAC’s serving proxy or proxies and the UAS’s serving proxy or proxies is to ignore the Answer-Mode header field. However, the serving proxies MAY exercise control over the requested answer mode, either inserting or deleting a Answer-Mode header field or altering the value of an existing header field in accord with local policy. Note that this may result in behavior that is inconsistent with user expectations, such as having a call that was intended to be a diagnostic loopback answered by a human, and consequently must be done very carefully. These serving proxies MAY also reject a request according to local policy, and SHOULD use the rejection codes as specified below for the UAS if they do so.

4.3 Procedures at the UAS

For a request having an Answer-Mode value of "Manual", the UAS SHOULD defer accepting the request until the user of the UAS has confirmed willingness to accept the request. This behavior MAY be altered as needed for unattended UAS or other local characteristics or policy. For example, an auto-attendant system that always answers automatically would go ahead and answer, despite the presence of the "Manual" Answer-Mode header field value.

For a request having an Answer-Mode value of "ManualReq", the UAS SHOULD defer accepting the request until the user of the UAS has confirmed willingness to accept the request. If the UAS is not capable of answering the request in this "Manual" mode or is unwilling to do so, it SHOULD reject the request with a "403 Forbidden" response and MAY include a Reason [7] header field value of:

Reason: SIP ;cause=403 ;text="manual answer forbidden"

For a request having an Answer-Mode value of "Auto", the UAS SHOULD, if the calling party is authenticated and authorized for automatic answering, accept the request without further user input. The UAS MAY, according to local policy or user preferences, treat this request as it would treat a request having a Answer-Mode with a value of "Manual" or having no Answer-Mode header field. If the calling party is not authenticated and authorized for automatic answer, the UAS may either handle the request as per "manual", or reject the
request. If the UAS rejects the request, it SHOULD do so with a "403 Forbidden" response, and MAY include a Reason [7] header field value of:

Reason: SIP ;cause=403 ;text="automatic answer forbidden"

For a request having an Answer-Mode value of "AutoReq", the UAS SHOULD apply authentication and authorization checks before accepting such a request. The UAS MUST NOT allow "manual" answer of this request, but MAY reject it. If, for whatever reason, the UAS chooses not to accept the request automatically, the UAS MUST reject the request and SHOULD do so with a "403 Forbidden" response, and MAY include a Reason [7] header field value of:

Reason: SIP ;cause=403 ;text="automatic answer forbidden"

4.4 Issues with Automatic Answering and Forking

One of the well-known issues with forking is the problem of multiple acceptance. If an INVITE request is forked to several UAS, and more than one of those UAS respond with a 200 OK, the conventional approach is to continue the dialog with the first respondent, and tear down the dialog (via BYE) with all other respondents.

While this problem exists without an auto-answer negotiation capability, it is apparent that widespread adoption of UAS that engage in auto-answer behavior will exacerbate the multiple acceptance problem. Consequently, systems designers need to take this aspect into consideration. In general, auto-answer is probably NOT RECOMMENDED in environments that include forking.

As an alternative, it might be reasonable to use a variation on manual-answer combined with no alerting and early media. In this approach, the initial message or talk-burst is transmitted as early media to all recipients, where it is displayed or played out. Any response utterance from the user of a UAS following this would serve as an "acceptance", resulting in a 200 OK response being transmitted by their UAS. Consequently, the race-condition for acceptance would be limited to the subset of UAs actually responding under user control, rather than the full set of UAS to which the request was forked.

Another alternative would be to use dynamic conferencing instead of forking. In this approach, instead of forking the request, a conference would be initiated and all UAs invited into that conference. The mixer attached to the conference would then mediate traffic flows appropriately.
5. Usage of the Alert-Mode Header Field, Option, and Media Feature Tags
   In a Request

   The Alert-Mode header field is used by a UAC to request specific
   handling of an INVITE request by the responding UAS related to the
   alerting of the user of the UAS. If no Alert-Mode header field is
   included in the request, alerting behavior is at the discretion of
   the UAS, as it would be in the absence of this specification. The
   desired handling is indicated by the value of the Alert-Mode
   header field, as follows:

   Normal: The UAS is asked to treat the request as it normally would in
   the absence of this specification and exercise whatever alerting
   mechanism it might have and be configured to use.
   Null: The UAS is asked to not alert its user to the request.

5.1 Procedures at the UAC

5.1.1 All Requests

   A UAC supporting this specification indicates its support for this
   extension by including an option tag of "answermode" in the Supported
   header field of all requests it sends.

5.1.2 REGISTER Transactions

   To indicate that it supports the alert-mode negotiation feature, a UA
   includes a SIP extension feature tag of "alertmode" in the Contact:
   header field of its REGISTER requests. This usage of feature tags is
   described in [5].

5.1.3 INVITE transactions

   A UAC supporting this specification includes a Alert-Mode header
   field and appropriate value in an INVITE where it wishes to influence
   the alerting mode of the responding UAS.

   To request that the UAS not alert its user the UAC includes a Alert-
   Mode header field having a value of "Null".

   To request that the UAS apply its normal procedures for alerting the
   user the UAC either includes a Alert-Mode header field having a value
   of "Normal" or it includes no Alert-Mode header field.

   To require that the UAS either support this extension or reject the
   request, the UAC includes a Required: header field having a value of
   "alertmode".
5.2 Procedures at Intermediate Proxies

The general procedure at all intermediate proxies including the UAC’s serving proxy or proxies and the UAS’s serving proxy or proxies is to ignore the Alert-Mode header field. However, the serving proxies MAY exercise control over the requested answer mode, either inserting or deleting a Alert-Mode header field or altering the value of an existing header field in accord with local policy. Note that this may result in behavior that is inconsistent with user expectations, such as having a call that was intended to be a silent diagnostic loopback answered by a human, and consequently must be done very carefully. These serving proxies MAY also reject a request according to local policy, and SHOULD use the rejection codes as specified below for the UAS if they do so.

5.3 Procedures at the UAS

A UAS supporting this specification considers the value of the Alert-Mode header field in an INVITE request in determining how and/or whether to alert the user of the UAS to the request. The UAS may also consider local policy, the presence of an authenticated identity or other authentication, and other elements of the request in making this determination.

If the conclusion is to alert the user, the UAS invokes its preferred alerting mechanism. If the conclusion is to not alert the user, the UAS proceeds to process the request. Note that the decision of whether to accept the request is independent of the alerting decision, but one can generally not expect the user to make this decision unless the user has been alerted to the request.

The general intent of a request having a Alert-Mode header field with a value of "Null" is that the user not be invasively interrupted by the request. Consequently, it might be appropriate to invoke a less-disruptive alerting mechanism (perhaps blinking a small light) as an alternative to not invoking any alerting mechanism.

6. Usage of the Answer-Mode Header Field in a Response

The Answer-Mode header field may be inserted by a UAS into a response in order to indicate how it handled the associated request with respect to automatic answering functionality. The UAC may use this information to inform the user or otherwise adapt the behavior of the user interface. The handling is indicated by the the value of the Answer-Mode header field, as follows:
Manual: The UAS responded after the user of the UAS interacted with the user interface (UI) of the UAS in such a way as to indicate that the user desires the UAS to accept the request.

ManualReq: The UAS responded manually, as above. Further, the request contained a Answer-Mode header field with the value "ManualReq", and the UAS has honored this requirement.

Auto: The UAS responded automatically, without waiting for the user of the UAS to interact with the UI of the UAS in such a way as to indicate that the user desires the UAS to accept the request.

AutoReq: The UAS responded automatically (as above). Further, the request contained a Answer-Mode header field with the value "AutoReq", and the UAS has honored this requirement.

The Answer-Mode header field, when used in a response, is only valid in a 200 OK response to an INVITE request.

6.1 Procedures at the UAS

A UAS supporting this specification inserts a Answer-Mode header field into the 200 OK response to an INVITE request when it wishes to inform the UAC as to whether the request was answered manually or automatically. The full rationale for including or not including this header field in a response is outside of the scope of this specification. However, it is reasonable for a UAS to assume that if the UAC included a Answer-Mode header field in the request that it would probably like to see a Answer-Mode header field in the response.

6.2 Procedures at the UAC

A UAC can use the value of the Answer-Mode header field, if present, to adapt the user interface and/or inform the user about the handling of the request. For example, the user of a push-to-talk system might speak differently if she knows that the called party answered "in person" vs. having the call blare out of an unattended speaker phone.

7. Examples of Usage

The following examples show Bob registering a contact that supports negotiation of answer mode and alerting mode. Alice then calls Bob with an INVITE request, asking for automatic answering with normal alerting and explicitly asking that the request not be routed to contacts that have not indicated support for this extension. Further, Alice requires that the request be rejected if Bob’s UA does not support negotiation of alerting and answer modes. Bob responds with a 200 OK indicating that the call was answered automatically.
7.1 REGISTER Request

```
REGISTER sip:example.com SIP/2.0
From: Bob <sip:bob@example.com>
To: Bob <sip:bob@example.com>
Contact: sip:cell-phone@example.com;
+sip.extensions="answermode";
methods="INVITE,BYE,OPTIONS,CANCEL,ACK"
```

7.2 INVITE Request

```
INVITE <sip:bob@example.com SIP/2.0>
Via: SIP/2.0/TCP client-alice.example.com:5060;branch=z9hG4bK74b43
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@example.com>
Call-ID:3848276298220188511@client-alice.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Requires: answermode, alertmode
Accept-contact:*;require;explicit;
+sip.extensions="answermode";
Answer-Mode: Auto
Alert-Mode: Null
Content-Type: application/sdp
Content-Length: ...
```

7.3 200 OK response

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP client-alice.example.com:5060;branch=z9hG4bK74bf9
From: Alice <sip:alice@example.com>;tag=9fxced76sl
To: Bob <sip:bob@example.com>;tag=8321234356
Call-ID: 3848276298220188511@client-alice.example.com
CSeq: 1 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Answer-Mode: Auto
Content-Type: application/sdp
Content-Length: ...
```

8. Security Considerations

This specification adds the ability for a UAC to request potentially risky user interface behavior relating to the acceptance of an INVITE request by the UAS receiving the request. These behaviors include accepting the request without notification of the user of the UAS, and accepting the request without input to the UAS by the user of the
UAS.

There are several attacks possible here, with the most obvious being the ability to turn a phone into a remote listening device without its user being aware of it. Additional attacks include reverse charge fraud, unsolicited "push to talk" communications (SPPTT), battery-rundown denial-of-service, "forced busy" denial of service, and phishing via session insertion (where an ongoing session is replaced by another without the victim’s awareness.

In the most common use cases, the security aspects are somewhat mitigated by design aspects of the application. For example, in push-to-talk applications, no media is sent from the called UA without user input (the "push" of "push-to-talk"). Consequently, there is no "bugging" attack when the "Null" Alert-Mode option is exercised in conjunction with automatic answering. Furthermore, the incoming initial talk burst, if present, may serve to alert the called user. However, there is still the potential for an "unsolicited message transmission". For example, the initial talk-burst of an auto-answered push-to-talk session might include an advertisement for pharmaceuticals, or broadcast rude noises in the tradition of the "whoopie cushion."

Consequently, the UAS generally or its supporting proxy MUST authenticate the sender of such requests, using mechanisms such as SIP Digest Authentication, [2], the SIP Identity mechanism [13], or the SIP mechanism for Asserted Identity Within Private Networks[8], in networks for which it is suitable.

The authenticated identity of the requester MUST then be matched against authorization policy appropriate to the requested application. For example, it might be appropriate to allow a designated systems administrator to start a diagnostic loopback session without alerting the user. It might also be appropriate to allow a known "buddy" to start a push-to-talk session without requiring the user of the UAS to actively accept the call. It is almost certainly NOT appropriate to allow an unauthenticated and unauthorized requester to start a session without alerting and receiving a confirmation of acceptance (manual answer) from the targeted user.

9. IANA Considerations

9.1 Registration of Header Fields

This document defines new SIP header fields named "Answer-Mode", "Alert-Mode", and "Answer-Mode".
The following rows shall be added to the "Header Fields" section of the SIP parameter registry:

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Compact Form</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer-Mode</td>
<td></td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Alert-Mode</td>
<td></td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Answer-Mode</td>
<td></td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Editor Note: [RFCXXXX] should be replaced with the designation of this document.

9.2 Registration of Header Field Parameters

This document defines parameters for the header fields defined in the preceding section. The header field named "Answer-Mode" may take the values "Manual", "Auto", or "AutoReq". The header field named "Alert-Mode" may take the values "Normal" or "Null".

The following rows shall be added to the "Header Field Parameters and Parameter Values" section of the SIP parameter registry:

<table>
<thead>
<tr>
<th>Header Field</th>
<th>Parameter Name</th>
<th>Predefined Values</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer-Mode</td>
<td>Manual</td>
<td>Yes</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Answer-Mode</td>
<td>Auto</td>
<td>Yes</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Answer-Mode</td>
<td>AutoReq</td>
<td>Yes</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Alert-Mode</td>
<td>Normal</td>
<td>Yes</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Alert-Mode</td>
<td>Null</td>
<td>Yes</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Editor Note: [RFCXXXX] should be replaced with the designation of this document.

9.3 Registration of Extension Option Tags

This document defines new SIP option tags "answermode" and "alertmode".
The following rows shall be added to the "Option Tags" section of the SIP Parameters registry:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>answermode</td>
<td>This option tag is used in a Requires header field to indicate that the UAS must support negotiation of answer mode.</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>alertmode</td>
<td>This option tag is used in a Requires header field to indicate that the UAS must support negotiation of alerting mode.</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Editor Note: [RFCXXXX] should be replaced with the designation of this document.

10. Acknowledgements

This document draws requirements and a large part of its methodology from the work of the Open Mobile Alliance, and specifically from the internet draft [12] by Andrew Allen, Jan Holm, and Tom Hallin.

The editor would also like to recognize the contributions of David Oran and others who argued on the SIPPING mailing list and at the OMA ad-hoc meeting at IETF 62 that the underlying ideas of the above draft were broadly applicable to the SIP community, and that the concepts of alerting and answering should be clearly delineated.

11. References

11.1 Normative References


11.2 Informative References


Authors’ Addresses

Dean Willis (editor)
Cisco Systems
3100 Independence Pkwy #311-164
Plano, Texas  75075
USA

Phone: unlisted
Fax: unlisted
Email: dean.willis@softarmor.com

Andrew Allen
Research in Motion (RIM)
122 West John Carpenter Parkway, Suite 430
Irving, Texas  75039
USA

Phone: unlisted
Fax: unlisted
Email: aallen@rim.com