RFC 3265bis: SIP Events Redux

Adam Roach

Anaheim, CA, USA

Tuesday, March 23, 2010

Changes from -00

- Clarified behavior when notifier cannot generate NOTIFY messages with the content type indicated in SUBSCRIBE.
- Changed timer name (now timer N)
- Clarified route establishment and record-route behavior
- Explained implications of state polling when state is not local to notifier
- Added information about targeting dialogs with subscription requests
- Added Call-Info to Table 2 (see Friday Discussion)

Open Issue: 202

- I posted proposed text to the list, and listened to the lovely sound of crickets.
- I'm just drawing attention to this here, take the actual discussion to the list.
- We *are not talking* about this today. The discussion in Stockholm clearly demonstrated that we need to hash out subtleties of text, not broad philosophical issues. That is best done in a text medium, not at microphones.
- Absent any mailing list discussion, the next version of the document will incorporate the text I posted on the list on March 2nd.

Open Issue: Allow-Events and Templates

- Also re-posted to the mailing list, also had no discussion.
- Proposal on mailing list (based on conversation in Stockholm) was to not include templates at all in Allow-Events, and let implementations probe for support
- Lacking any further mailing list discussion, the next version of the draft will incorporate this proposal.

Open Issue: Timer N and Re-SUBSCRIBEs

- Discussion (both in Stockholm and on mailing list) has been almost exclusively between me and James Polk.
- My interpretation of the most recent mailing list exchange is that James is okay with Timer N expiration being treated as other timeouts (i.e., it destroys the subscription)
- Need to add text to ensure proper handling with rate control (i.e., don't run Timer N if SUBSCRIBE response was 204).
- Lacking any further mailing list discussion, the next version of the draft will incorporate this handling.

Changes from RFC 3265 (1/3)

- Clarify handling of Route/Record-Route in NOTIFY
- Eliminate implicit subscriptions
- Deprecate dialog re-use
- Rationalize dialog creation
- Refactor behavior sections
- Clarify sections that need to be present in event packages
- Make CANCEL handling more explicit
- Remove State Agent Terminology

Changes from RFC 3265 (2/3)

- Bug 666: Clarify use of expires=xxx with terminated
- Bug 667: Reason code for unsub/poll not clearly spelled out
- Bug 669: Clarify: SUBSCRIBE for a duration might be answered with a NOTIFY/expires=0
- Bug 670: Dialog State Machine needs clarification
- Bug 671: Clarify timeout-based removal of subscriptions
- Bug 672: Mandate expires= in NOTIFY
- Bug 673: INVITE 481 response effect clarification
- Bug 677: SUBSCRIBE response matching text in error
- Bug 695: Document is not explicit about response to NOTIFY at subscription termination

Changes from RFC 3265 (3/3)

- Bug 696: Subscription state machine needs clarification
- Bug 697: Unsubscription behavior could be clarified
- Bug 699: NOTIFY and SUBSCRIBE are target refresh requests
- Bug 722: Inconsistent 423 reason phrase text
- Bug 741: guidance needed on when to not include Allow-Events
- Bug 744: 5xx to NOTIFY terminates a subscription, but should not
- Bug 752: Detection of forked requests is incorrect
- Bug 773: Reason code needs IANA registry
- Bug 774: Need new reason for terminating subscriptions to resources that never change

Protocol and Document Improvements

Application of Lessons Learned

Clarify handling of Route/ Record-Route in NOTIFY

Proxies need no additional behavior beyond that described in SIP [RFC3261] to support SUBSCRIBE and NOTIFY. If a proxy wishes to see all of the SUBSCRIBE and NOTIFY requests for a given dialog, it MUST add a Record-Route header field to the initial SUBSCRIBE request and all NOTIFY requests. It MAY choose to include Record-Route in subsequent SUBSCRIBE messages; however, these requests cannot cause the dialog's route set to be modified.

Eliminate Implicit Subscriptions

Notifiers learn about subscription requests by receiving SUBSCRIBE requests from interested parties. Notifiers MUST NOT create subscriptions except upon receipt of a SUBSCRIBE message. However, for historical reasons, the implicit creation of subscriptions as defined in [RFC3515] is still permitted.

Deprecate Dialog Re-use

- Moved all dialog re-use text into new section (4.5.2), which is used only for interoperation with legacy 3265 implementations.
- 3265bis *notifiers* (but not subscribers) are required to use GRUUs for local target (i.e., “Contact”)
  - This is not generally constraining, as most notifiers are full-fledged servers with DNS entries and/or static IP address. Creation of a “Self-Made” GRUU trivially involves adding a “;gr” to their normal contact URI.
- 3265bis *subscribers* must not attempt dialog re-use if the dialog’s remote contact is a GRUU.
- Added advisory text explaining how event packages can talk about existing dialogs (using, e.g., “Target-Dialog” header field)

Rationalize Dialog Creation

- Dialog creation now occurs when the NOTIFY arrives, rather than when the 200 occurs.
- This is easier to implement in subscriber implementations, and has no impact on notifiers.
- This change also does not impact proxy handling.

Refactor Behavior Sections

- Behavior description in 3265 was broken down along method lines (e.g., behavior for NOTIFY, behavior for SUBSCRIBE).
- Implementor feedback was that implementation of a subscriber or a notifier required a lot of skipping around.
- New document broken down along role (e.g., behavior for subscribers, behavior for notifiers).

Clarify sections that need to be present in event packages

This section covers several issues which should be taken into consideration when event packages based on SUBSCRIBE and NOTIFY are proposed. Event package definitions contain sections addressing each of these issues, ideally in the same order and with the same titles as the following sections.

Make CANCEL handling more explicit

Neither SUBSCRIBE nor NOTIFY messages can be canceled. If a UAS receives a CANCEL request that matches a known SUBSCRIBE or NOTIFY transaction, it MUST respond to the CANCEL request, but otherwise ignore it. In particular, the CANCEL request MUST NOT affect processing of the SUBSCRIBE or NOTIFY request in any way.

UACs SHOULD NOT send CANCEL requests for SUBSCRIBE or NOTIFY transactions.

Remove “State Agent” Terminology

- This is an editorial change that has no protocol impacts.
- Original vision in 3265 was that many event packages would use subscriptions that migrate between ephemeral endpoints and permanent servers.
- 3265 defined the term “State Agent” to help talk about these permanent servers and their role in subscription migration.
- In practice, migration of subscriptions is not used.
- Nearly all event packages defined to date get use this terminology incorrectly in their first draft. Most commonly, they get it backwards (claiming, for example, that State Agents are not used, when the event package relies on State Agents to work at all).
- 3265bis removes the term “State Agent.” The behaviors required to implement a State Agent (aggregation, migration, etc) remain in the draft.

Resolution of Reported Bugs

See http://bugs.sipit.net/

Bug 666: Clarify use of expires=xxx with terminated

If the "Subscription-State" value is "terminated", the subscriber should consider the subscription terminated. The "expires" parameter has no semantics for "terminated" -- notifiers SHOULD NOT include an "expires" parameter on a "Subscription-State" header field with a value of "terminated," and subscribers MUST ignore any such parameter, if present.

Bug 667: Reason code for unsub/ poll not clearly spelled out

timeout: The subscription has been terminated because it was not refreshed before it expired. Clients MAY re-subscribe immediately. The "retry-after" parameter has no semantics for "timeout". This reason code is also associated with polling of resource state, as detailed in Section 4.4.3.

Bug 669: Clarify: SUBSCRIBE for a duration might be answered with a NOTIFY/expires=0

The period of time for a subscription can be shortened to zero by the notifier. In other words, it is perfectly valid for a SUBSCRIBE with a non-zero expires to be answered with a NOTIFY that contains "Subscription-Status: terminated;reason=expired". This merely means that the notifier has shortened the subscription timeout to zero, and the subscription has expired instantaneously. The body may contain valid state, or it may contain a neutral state (see Section 5.4.7).

Bug 670: Dialog State Machine needs clarification

- Multiple usages deprecated, so the issue arises only during interoperation with legacy implementations.
- Additionally, the document now cites RFC 5057 for guidance regarding proper dialog handling under error circumstances

Bug 671: Clarify timeout-based removal of subscriptions

If the NOTIFY request fails due to expiration of SIP Timer F (transaction timeout), the notifier SHOULD remove the subscription.

Bug 672: Mandate expires= in NOTIFY

If the value of the "Subscription-State" header field is "active" or "pending", the notifier MUST also include in the "Subscription-State" header field an "expires" parameter which indicates the time remaining on the subscription.

(Was previously "…SHOULD also include…")

Bug 673: INVITE 481 response effect clarification

- Multiple usages deprecated, so the issue arises only during interoperation with legacy implementations.
- Additionally, the document now cites RFC 5057 for guidance regarding proper dialog handling under error circumstances

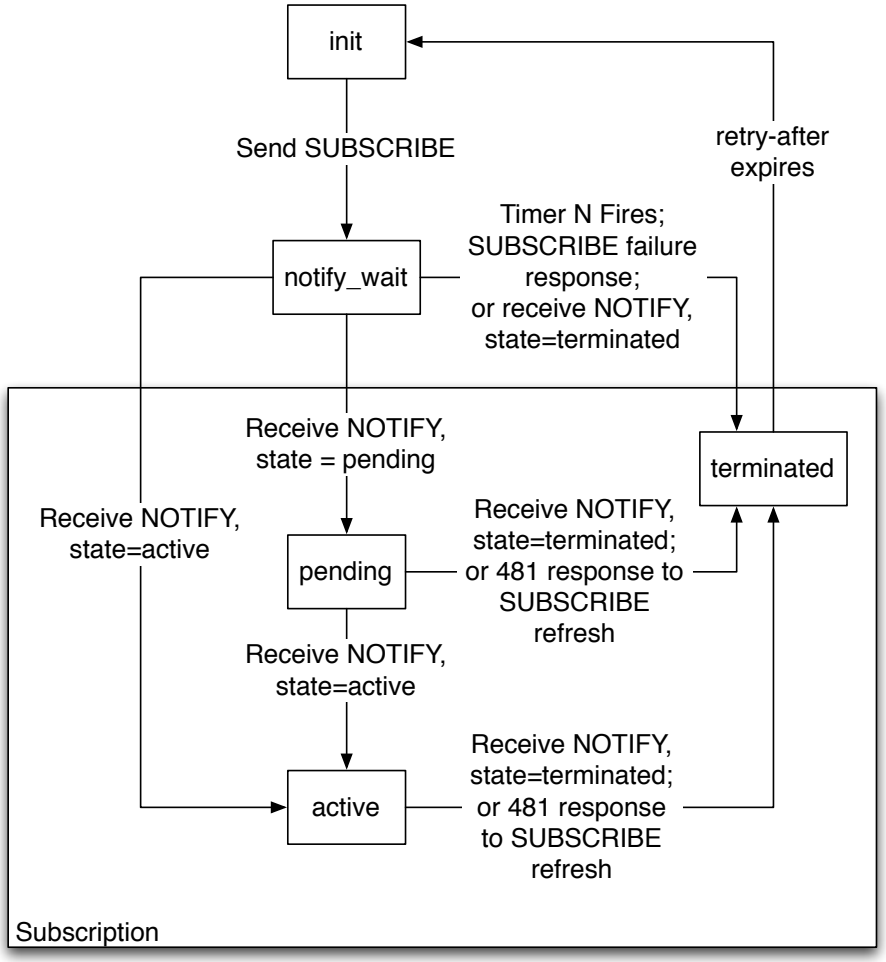
Bug 677: SUBSCRIBE response matching text in error

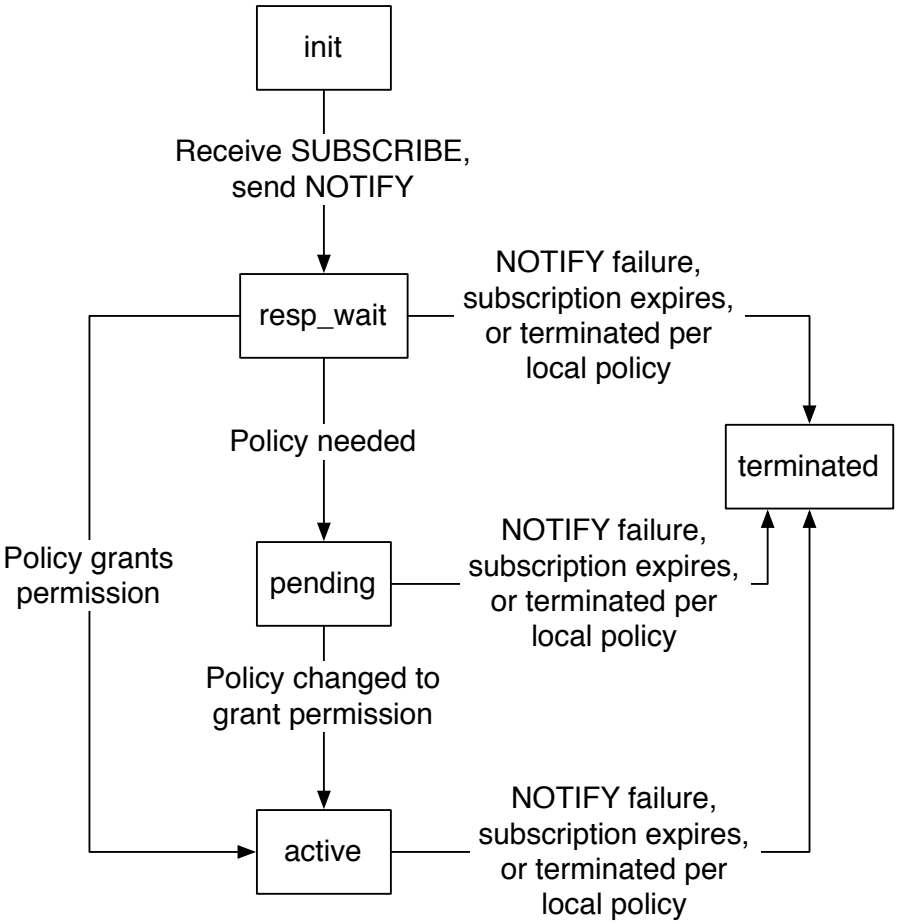
For the purposes of matching ~~responses and~~ NOTIFY messages with SUBSCRIBE messages, the event-type portion of the "Event" header is compared byte-by-byte and the "id" parameter token (if present) is compared byte- by-byte. An "Event" header field containing an "id" parameter never matches an "Event" header field without an "id" parameter. No other parameters are considered when performing a comparison. SUBSCRIBE responses are matched per the transaction handling rules in SIP [RFC3261].

Bug 695: Document is not explicit about response to NOTIFY at subscription termination

A subscription is destroyed after a notifier sends a NOTIFY request with a "Subscription-State" of "terminated." The subscriber will generally answer such final requests with a "200 OK" response (unless a condition warranting an alternate response has arisen).

Bug 696: Subscription state machine needs clarification





Bug 697: Unsubscription behavior could be clarified (1/2)

Clients can cause a subscription to be terminated immediately by sending a SUBSCRIBE with an "Expires" header field set to '0'. Notifiers largely treat this the same way as any other subscription expiration: they send a NOTIFY message containing a "Subscription- State" of "terminated", with a reason code of "timeout." For consistency with state polling (see Section 4.4.3) and refreshes, the notifier may choose to include resource state in this final NOTIFY. However, in some cases, including such state makes no sense. Under such circumstances, the notifier may choose to omit state information from the terminal NOTIFY message.

Bug 697: Unsubscription behavior could be clarified (2/2)

The final NOTIFY may or may not contain information about the state of the resource; subscribers need to be prepared to receive final NOTIFY messages both with and without state.

Bug 699: NOTIFY and SUBSCRIBE are target refresh requests

The SUBSCRIBE method is used to request current state and state updates from a remote node. SUBSCRIBE is a target refresh request, as that term is defined in SIP [RFC3261].

NOTIFY messages are sent to inform subscribers of changes in state to which the subscriber has a subscription. Subscriptions are typically put in place using the SUBSCRIBE method; however, it is possible that other means have been used. NOTIFY is a target refresh request, as that term is defined in SIP [RFC3261].

Bug 722: Inconsistent 423 reason phrase text

- Changed reason code to "Interval Too Brief" in Section 4.2.1.1 and Section 4.2.1.4, to match 423 reason code in SIP.

Bug 741: guidance needed on when to not include Allow-Events

(Non Normative): This information is very useful, for example, in allowing user agents to render particular interface elements appropriately according to whether the events required to implement the features they represent are supported by the appropriate nodes. On the other hand, it doesn't necessarily make much sense to indicate supported events inside a NOTIFY-established dialog if the only event package supported is the one associated with that subscription.

Bug 744: 5xx to NOTIFY terminates a subscription, but should not

If the NOTIFY transaction fails due to the receipt of a 404, 405, 410, 416, 480-485, 489, 501, or 604 response to the NOTIFY, the notifier MUST remove the corresponding subscription. See [RFC5057] for further details and notes about the effect of error codes on dialogs and usages within dialog (such as subscriptions).

Bug 752: Detection of forked requests is incorrect

If [forking] behavior is not allowed, the first potential dialog-establishing message will create a dialog. All subsequent NOTIFY messages which correspond to the SUBSCRIBE message (i.e., match "To", "From", "From" header "tag" parameter, "Call-ID", ~~"CSeq",~~ "Event", and "Event" header "id" parameter) but which do not match the dialog would be rejected with a 481 response.

Bug 773: Reason code needs IANA registry

- Added new section 7.2 to create Reason Code IANA registry
- Includes “badfilter” reason defined by RFC 4660 in initial registration list

Bug 774: Need new reason for terminating subscriptions to resources that never change

invariant: The subscription has been terminated because the resource state is guaranteed not to change for the foreseeable future. This may be the case, for example, when subscribing to the location information of a fixed-location land-line telephone. When using this reason code, notifiers are advised to include a "retry-after" parameter with a large value (for example, 31536000 -- or one year) to prevent older, RFC 3265-compliant clients from periodically resubscribing. Clients SHOULD NOT attempt to resubscribe after receiving a reason code of "invariant," regardless of the presence of or value of a "retry-after" parameter.