

# Outbound Changes and Issues

Rohan Mahy  
rohan@ekabal.com

# Agreed Changes since -04 (1/2)

- Moved STUN to a separate section. Reference this section from within the relevant sections in the rest of the document.
- Add language clarifying that UA MUST NOT send STUN without an explicit indication the server supports STUN.
- Add language describing that UA MUST stop sending STUN if it appears the server does not support it.
- Defined a 'sip-stun' option tag. UAs can optionally probe servers for it with OPTIONS. Clarified that UAs SHOULD NOT put this in a Proxy-Require. Explain that the first-hop MUST support this option-tag.
- Clarify that SIP/STUN in TLS is on the "inside". STUN used with Sigcomp-compressed SIP is "outside" the compression layer for UDP, but wrapped inside the well-known shim header for TCP-based transports.
- Change UAC MUST-->SHOULD register a flow for each member of outbound-proxy-set

# Agreed Changes since -04 (2/2)

- Reworded registrar and proxy in some places (introduce the term "Authoritative Proxy").
- Loosened restrictions on always storing a complete Path vector back to the registrar/authoritative proxy if a previous hop in the path vector is reachable.
- Added comment about reregistration typically happening over same flow as original registration.
- Restored sanity by restoring text which explains that registrations with the same reg-id replace the old registration.
- Added text about the 'ob' parameter which is used in Path header field URIs to make sure that the previous proxy that added a Path understood outbound processing. The registrar doesn't include Supported: outbound unless it could actually do outbound processing (ex: any Path headers have to have the 'ob' parameter).
- Added some text describing what a registration means when there is an instance-id, but no reg-id.

# Late change: New response code

- We tried 410 Gone. Wrong:
  - Adam sends Cullen a mid-dialog request when one flow fails. The proxy forwards the 410 as the best/final response. Adam thinks “oh, Cullen ceased to exist”.
- We tried 480 Temporarily Unavailable. Wrong:
  - Rohan puts his phone in Do Not Disturb mode. Someone calls Rohan. The authoritative proxy tries several forks towards the phone. The proxy interprets the 480 response as flow failures and deletes ALL flows to the phone.
- Added new response code:
  - ~~489~~ 430 Flow Failed
  - Means exactly what it says.

# Late addition: stable flow timer

- Problem: Don't want flows that aren't stable to skip our avalanche restart timers
- In draft:
  - Flow has to be up for a short period of time before considered stable
  - If a flow is stable, the UA retries immediately after a flow failure
  - If UA registered on flow but flow is unstable, the UA waits as if the registration failed
  - Default value is 120 seconds

# Bug A: need to mention rport

- rport parameter is required for outbound-initiated flows to work through a NAT or Firewall (sort of the whole point)
- Will add text explaining that if the UA wants outbound to work through NATs/FWs it needs to put rport in its Via and send from the port it is prepared to receive on.

# Issue B: Verify outbound support on first hop or all hops?

- Current text (-05) requires each proxy to verify the previous element in the Path vector is reachable.
- Each proxy that adds a Path URI needs to add an 'ob' parameter otherwise the registrar reverts to non-outbound behavior
- Probably only need first hop and the registrar to support outbound. But this assumption is brittle.
- What do we want to do?
  - Leave spec as-is (each element in Path needs to support outbound)
  - Only require first hop and registrar

# Issue C: max-flows parameter

- I assumed that setting a max-flows parameter was out of scope.
  - We have no explicit requirements for this
  - Max number of flows already limited by the number of URIs in the outbound-proxy-set
- Visited network could specify more URIs in outbound-proxy-set than the Home registrar is prepared to handle.
  - Suggestion from Christer: return a max-flows header/parameter in registration response to further limit number of flows.
  - What response code would be used to refuse additional flow registrations?
- Proposal: Defer

# Issue D: First EP supports outbound, subsequent do not

- UA registers first flow
  - send register with reg-id=1 to EP1
  - gets back Supported: outbound
- If EP2 doesn't support outbound, weird things can happen.
- Note in doc: UA can register subsequent flows (ex: to EP2) that have a reg-id with Require: outbound

# Issue E: First EP doesn't support outbound, subsequent EP does.

- Corollary to Issue D
- If first flow to EP1 does not support outbound, subsequent flows (EP2) probably shouldn't contain a reg-id.

# Issue F: Detecting instance-id binding rules

- Only in case where UA registers with instance-id but no reg-id. Can't tell if registrar used old (3261) binding rules or new instance-id binding rules.
- Used to be easy. Used to be defined in GRUU—you got back a gruu parameter
- Does the UA really need to know?
- What can we do?
  - Could add a new 'instance-id' option-tag
  - Don't make any change; don't worry about it
- Proposal: Don't worry. Fix later if we need to.

# Issue 3G: Binding behavior without flow-tokens

- 3GPP wants to solve several requirements that happen to be solved by outbound.
- They *don't* want the flow-token behavior that is applicable to generic UAs on the public Internet
- What's similar:
  - 3GPP and others want to store multiple path vectors back to an instance, each associated with a reg-id.
  - New registrations with the same reg-id would replace the old binding.
- BUT they want to do this unrelated to outbound flow-token processing.
- They want separation of binding behavior and flow-token behavior
- Why? Their IPsec UDP uses several pairs of actual flows, instead of just one.

# Issue 3G: What can we do?

- 1) Leave current draft as is
- 2) relax flow-token language slightly
  - instead of flow-token saving specific UDP address/port tuples over which the request arrived, make language fuzzy to “save token which points to a ‘logical flow’ that is known to deliver data that specific UA instance”
- 3) replace reg-id with two new parameters
  - ‘path-id’ replaces reg-id. Means I want this path to the instance stored
  - ‘ob’ in a Contact means UA wants outbound flow-token behavior.