SIP Session Timer Glare Handling

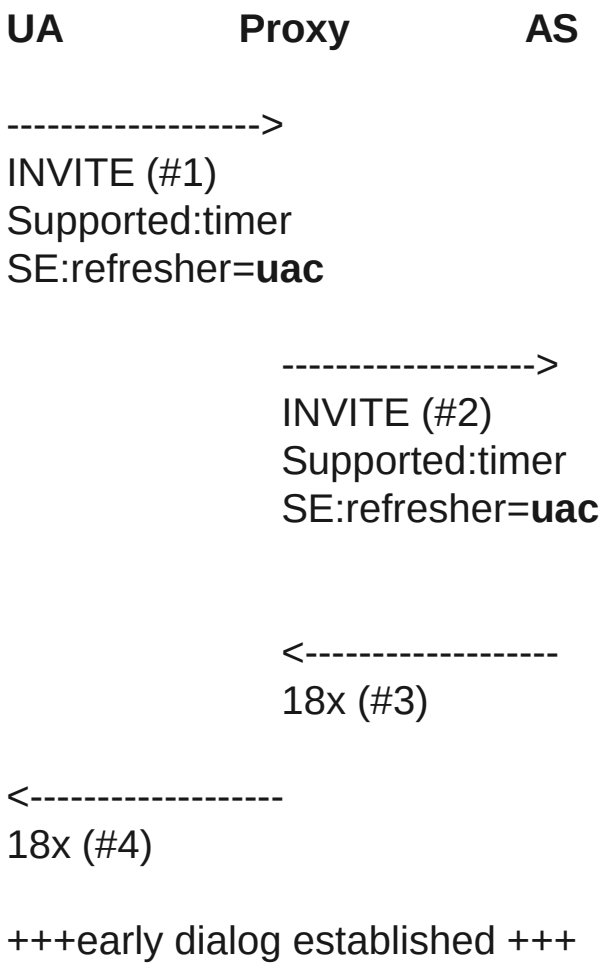
IETF#101
London, UK
draft-ietf-sipcore-sessiontimer-race-01

Christer.Holmberg@ericsson.com

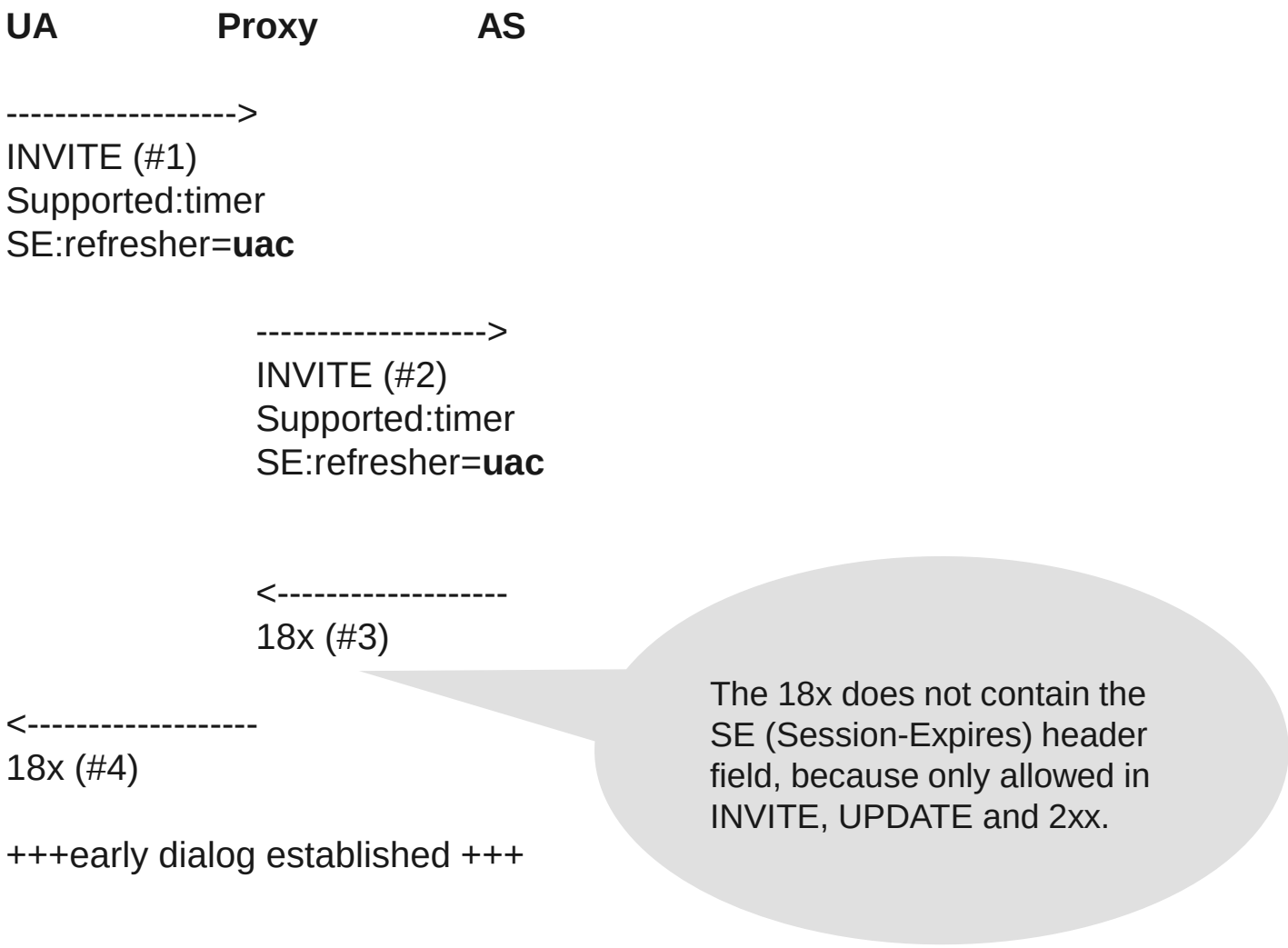
THE AGENDA

- **THE FLOW**
  - Real-life deployment use-case causing problem
- **THE BLAME**
  - Different opinions on which entity to blame
- **THE STANDARD**
  - Ambiguities/missing parts in RFC 4028
- **THE SOLUTION**
  - Describing the solution currently defined in the draft
- **THE LIST**
  - Picks from the mailing list comments on the defined solution
- **THE WAY FORWARD**

THE FLOW


UA
Proxy
AS
---------->
INVITE (#1)
Supported:timer
SE:refresher=uac
---------->
INVITE (#2)
Supported:timer
SE:refresher=uac
<----------
18x (#3)
<----------
18x (#4)
+++early dialog established +++

THE FLOW


UA
Proxy
AS
---------->
INVITE (#1)
Supported:timer
SE:refresher=uac
---------->
INVITE (#2)
Supported:timer
SE:refresher=uac
<----------
18x (#3)
<----------
18x (#4)
+++early dialog established +++
The 18x does not contain the SE (Session-Expires) header field, because only allowed in INVITE, UPDATE and 2xx.

THE FLOW

UA **Proxy** **AS**

<----------
UPDATE (#5)
Supported:timer
SE:refresher=**uas**

<----------
UPDATE (#6)
Supported:timer
SE:refresher=**uas**

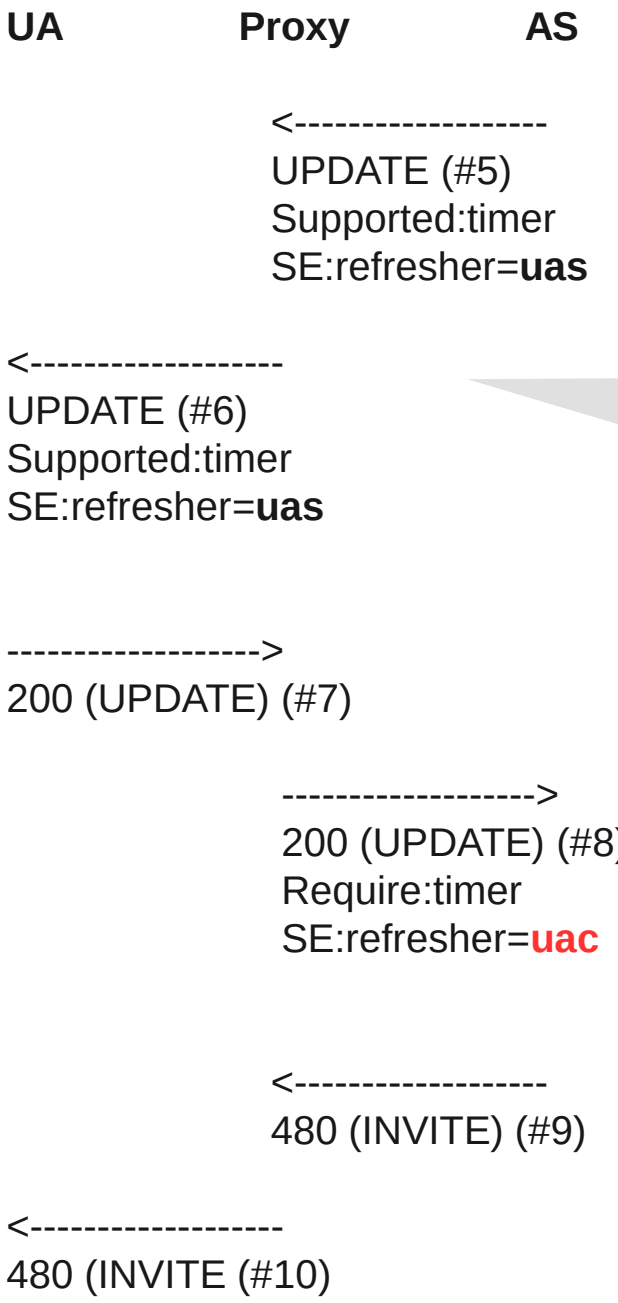
---------->
200 (UPDATE) (#7)

---------->
200 (UPDATE) (#8)
Require:timer
SE:refresher=**uac**

<----------
480 (INVITE) (#9)

<----------
480 (INVITE (#10)

THE FLOW

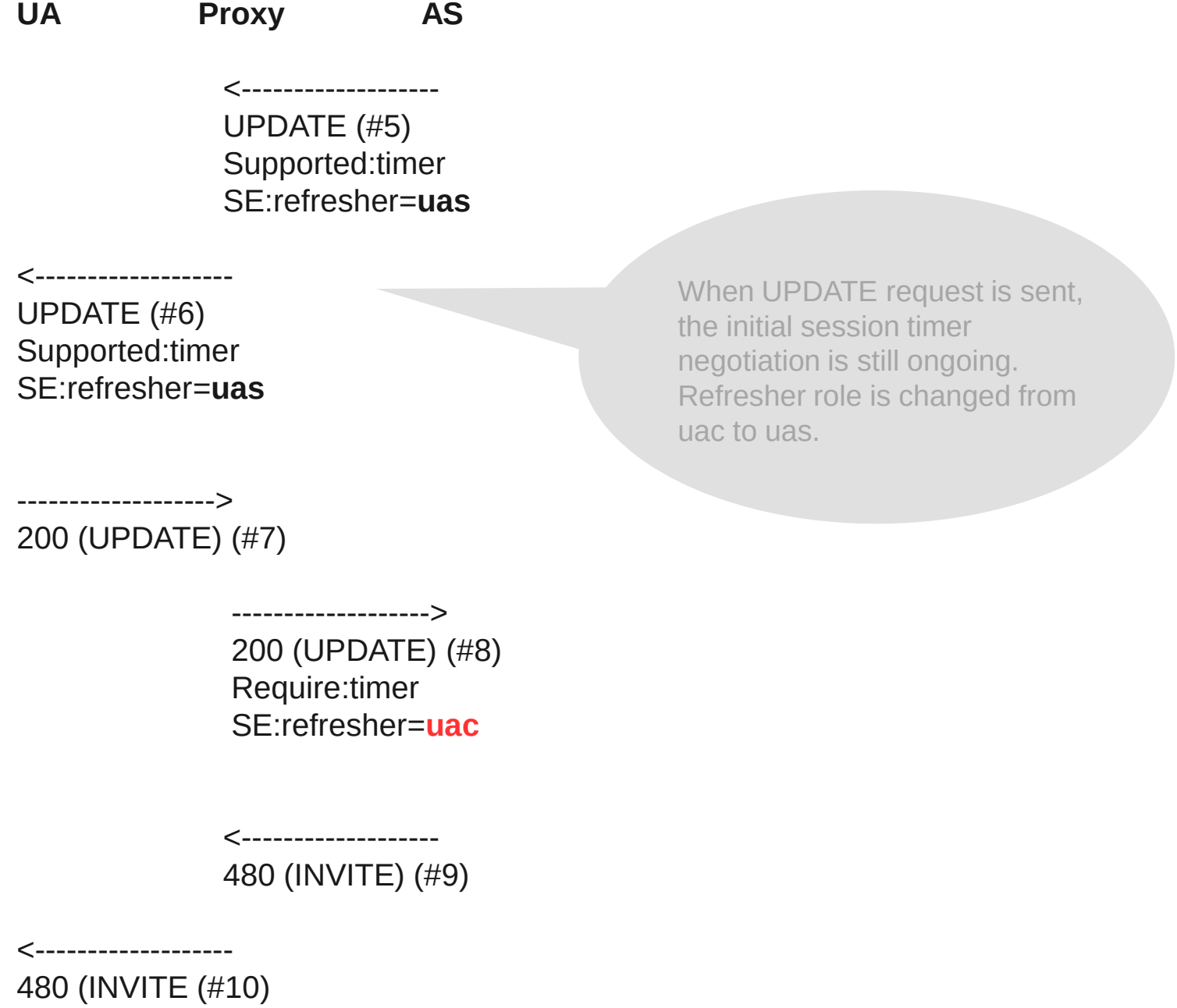


When UPDATE request is sent, the initial session timer negotiation is still ongoing. Refresher role is changed from uac to uas.

THE FLOW

"In a session refresh request sent within a dialog **with an active session timer**, the Session-Expires header field SHOULD be present.“

Q: Is session timer “active”?

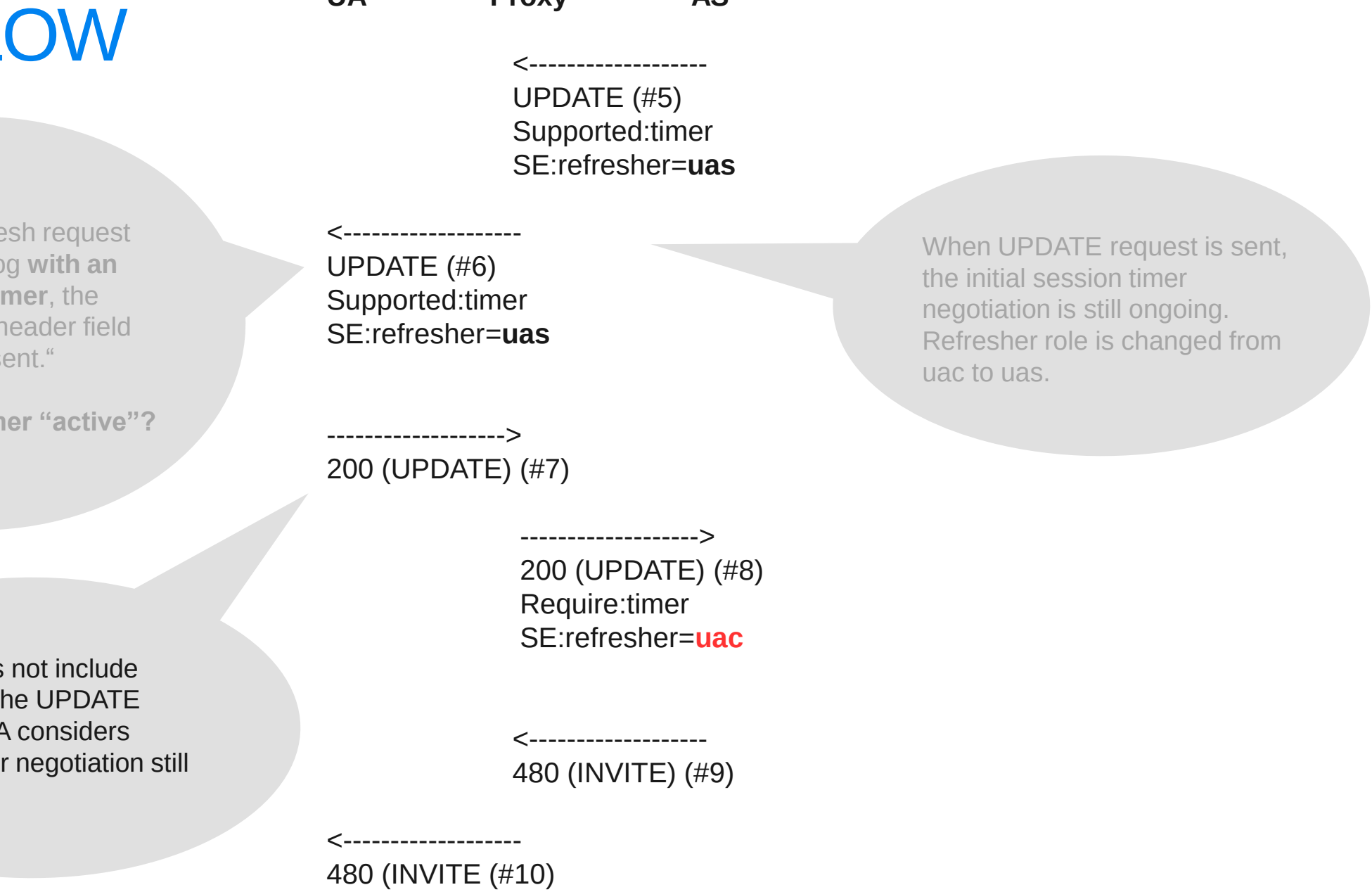


When UPDATE request is sent, the initial session timer negotiation is still ongoing. Refresher role is changed from uac to uas.

THE FLOW

"In a session refresh request sent within a dialog **with an active session timer**, the Session-Expires header field SHOULD be present."

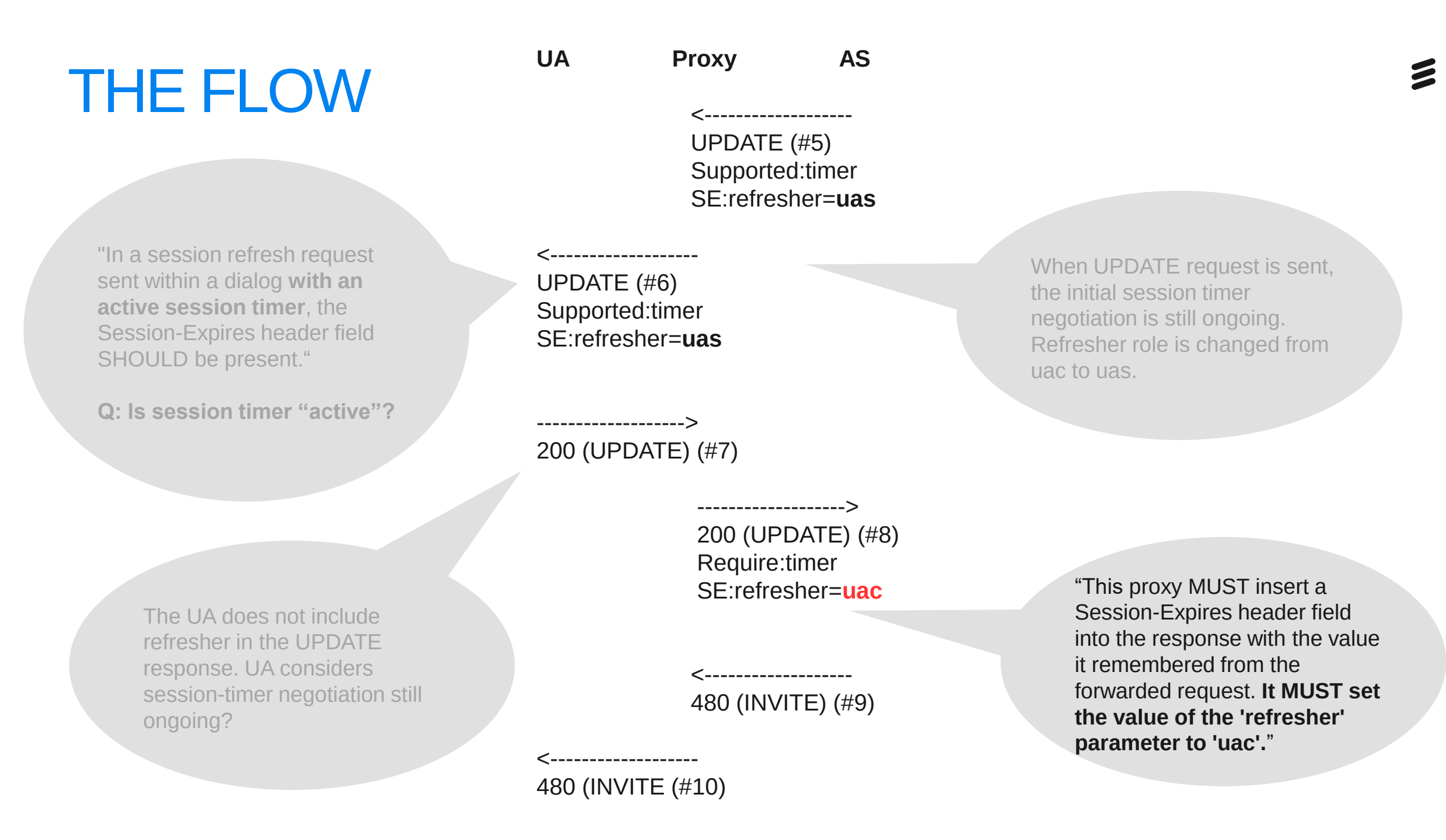
Q: Is session timer "active"?



When UPDATE request is sent, the initial session timer negotiation is still ongoing. Refresher role is changed from uac to uas.

The UA does not include refresher in the UPDATE response. UA considers session-timer negotiation still ongoing?

THE FLOW



"In a session refresh request sent within a dialog **with an active session timer**, the Session-Expires header field SHOULD be present."

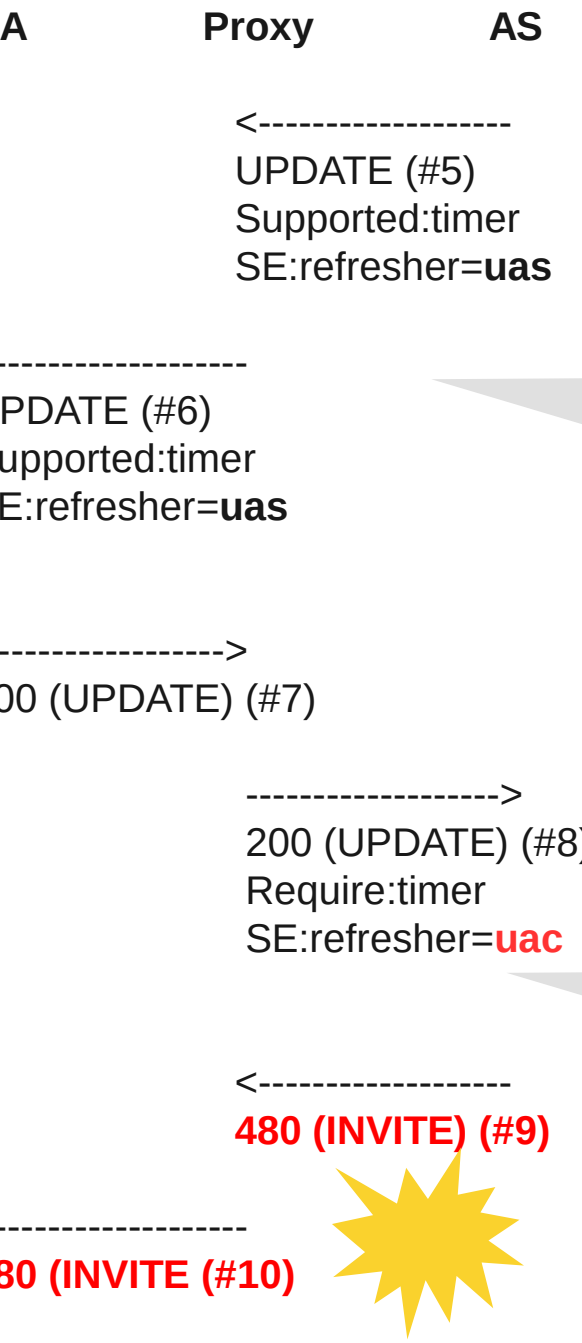
Q: Is session timer "active"?

When UPDATE request is sent, the initial session timer negotiation is still ongoing. Refresher role is changed from uac to uas.

The UA does not include refresher in the UPDATE response. UA considers session-timer negotiation still ongoing?

"This proxy MUST insert a Session-Expires header field into the response with the value it remembered from the forwarded request. **It MUST set the value of the 'refresher' parameter to 'uac'.**"

THE FLOW



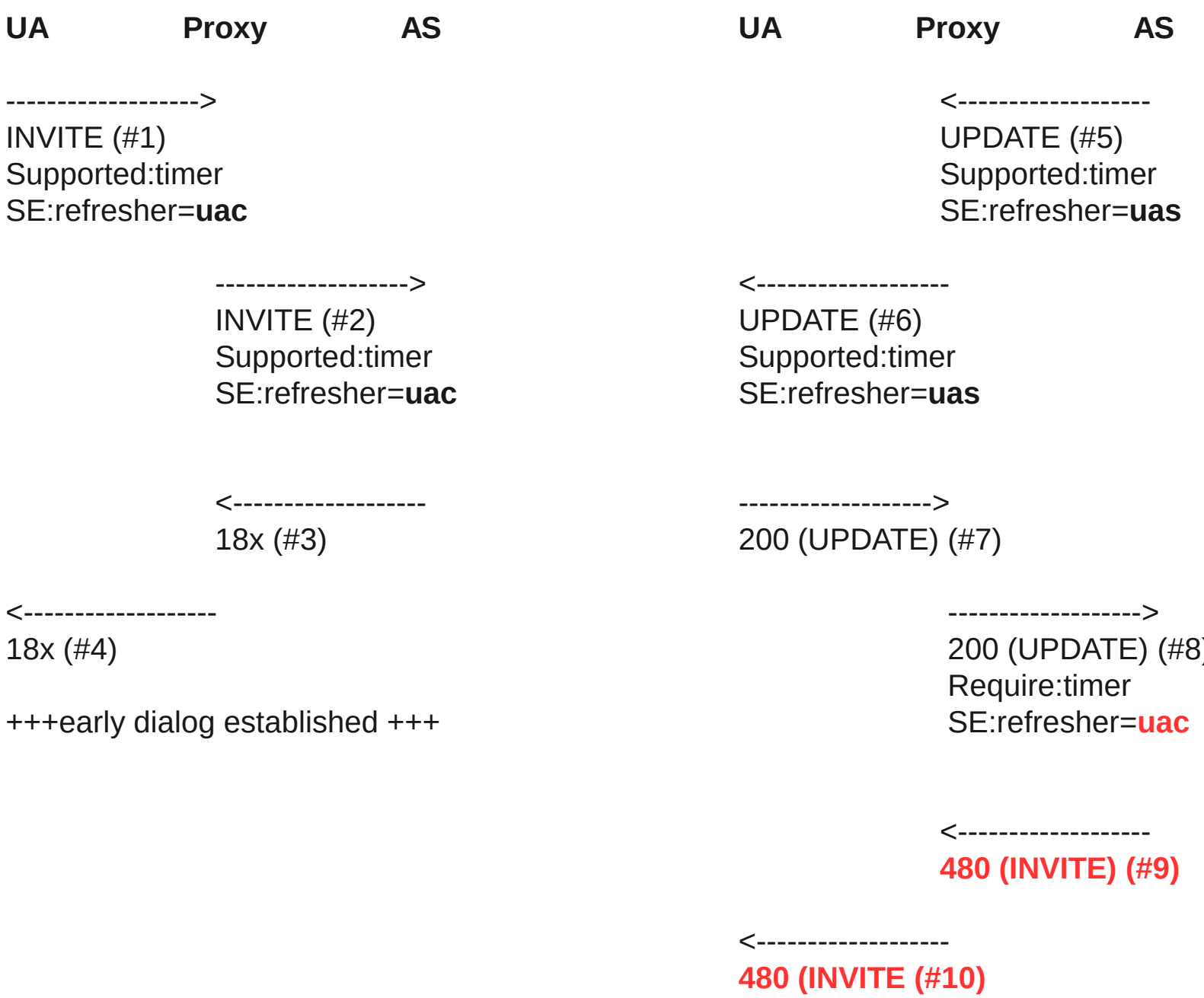
"In a session refresh request sent within a dialog **with an active session timer**, the Session-Expires header field SHOULD be present."

Q: Is session timer "active"?

When UPDATE request is sent, the initial session timer negotiation is still ongoing. Refresher role is changed from uac to uas.

The UA does not include refresher in the UPDATE response. UA considers session-timer negotiation still ongoing?

"This proxy MUST insert a Session-Expires header field into the response with the value it remembered from the forwarded request. **It MUST set the value of the 'refresher' parameter to 'uac'.**"


UA
Proxy
AS
---------->
INVITE (#1)
Supported:timer
SE:refresher=uac
---------->
INVITE (#2)
Supported:timer
SE:refresher=uac
<----------
18x (#3)
<----------
18x (#4)
+++early dialog established +++
<----------
UPDATE (#5)
Supported:timer
SE:refresher=uas
<----------
UPDATE (#6)
Supported:timer
SE:refresher=uas
---------->
200 (UPDATE) (#7)
---------->
200 (UPDATE) (#8)
Require:timer
SE:refresher=uac
<----------
480 (INVITE) (#9)
<----------
480 (INVITE (#10)

THE BLAME

- Application Server (AS) shall not change refresher role while initial session-timer negotiation is still ongoing.
- User Agent (UA) shall include the refresher parameter in the UPDATE response.
- Application Server (AS) shall not reject the INVITE request because of the session-timer ambiguity.
- Etc etc etc.

THE STANDARD

- RFC 4028 does describe session-timer procedures for mid-dialog requests
  - However, the procedures do not cover the case when a mid-dialog request is sent during an ongoing session-timer negotiation transaction (INVITE transaction).

THE SOLUTION (DRAFT)

- A Session-Expires header field can only be included in a session refresh request if there is no ongoing negotiation of usage of the session timer mechanism, and if there is no ongoing INVITE transaction.

- A UA shall, if it receives a session refresh request with a Session-Expires header field during an ongoing negotiation of usage of the session timer mechanism, or when there is an ongoing INVITE transaction, send a 491 (Request Pending) response to the request.

- The absence of a Session-Expires header field in a response will disable usage of the session timer mechanism only if the associated request contained a Session-Expires header field.

THE COMMENTS

— "*I'll propose that the session timer negotiation never be done with an UPDATE within an INVITE transaction. (Regardless of whether than INVITE is negotiating a session timer or not.) I think this resolves the problem that you have encountered.*" (Paul K, 09-13-2017)

  — "*I had a chat with some product people, and they said that there actually ARE cases where the session timer is negotiated using UPDATE when the initial INVITE transaction is still ongoing. There are cases where the INVITE only contains Supported:timer, but the actual negotiation is done using UPDATE.*" (Christer H, 09-14-2017)

THE COMMENTS

- “*I am inclined to keep things simple by saying that the 2xx response to every INVITE or UPDATE redefines the state of the session timer, either on or off. That will of course break the use case you describe above.*“ (Paul K, 09-14-2017)
  - “*The problem here is that the UAC sends INVITE with S-E, and then receives an UPDATE with S-E before it has received a 2xx response to the INVITE - making the UAC think that there is a S-E glare situation (i.e., the UAS also tries to initiate session timer negotiation).*” (Christer H, 10-05-2017)

THE WAY FORWARD

- Whatever solution we decide upon is going to impact existing deployments
- Aim should be to minimize impact on user endpoints
  - Network entities easier to fix
- Aim is to fix/clarify existing session-timer mechanism, not re-define it from scratch

THE END