

SCTP Negotiation Acceleration Protocol

draft-hancke-tsvwg-snap-00

Philipp Hancke
IETF 125 Shenzhen
18 March 2026



draft-hancke-tsvwg-snap-00

- Mostly a SCTP topic
 - but the proposed solution is based on SDP
 - What working group is the right place?
- WebRTC uses SCTP for datachannels
 - DCEP, RFC 8832
 - runs over DTLS after ICE
- Time to open datachannel can be reduced
 - ICE/DTLS (SPED, see other slides)
 - Datachannel (SNAP)
 - DCEP already allows sending data without waiting for ack (and negotiated channels exist)

draft-hancke-tsvwg-snap-00

```
Offerer                                     Answerer
|----- SDP Offer ----->|
|<-1----- SDP Answer -----|
|
|<-2----- ICE/Connectivity Checks ----->|
|
|<----- DTLS ClientHello -----|
|--3----- DTLS ServerHello ----->|
|<----- DTLS Finished -----|
|--4----- DTLS Finished ----->|
|
|----- SCTP INIT ----->|
|<-5----- SCTP INIT ACK -----|
|----- SCTP COOKIE ECHO ----->|
|<-6----- SCTP COOKIE ACK -----|
|          OFFERER DATA READY (6 RTT)      |
```

```
Offerer                                     Answerer
|----- SDP Offer (sctp-init) ----->|
|<-1----- SDP Answer (sctp-init) -----|
|
|<-2----- ICE/Connectivity Checks ----->|
|
|<----- DTLS ClientHello -----|
|--3----- DTLS ServerHello ----->|
|<----- DTLS Finished -----|
|--4----- DTLS Finished ----->|
|          OFFERER DATA READY (4 RTT)      |
```

- Left: WebRTC 1.0
- Right: WebRTC with SNAP
- DTLS 1.2 with actpass offer/active answer in both cases

draft-hancke-tsvwg-snap-00

- SCTP exchanges INIT/INIT-ACK + COOKIE/COOKIE-ACK
 - 1-2 RTT without packet loss (1 RTT in theory, 2 RTT observed)
 - In WebRTC we do this after SDP exchange and after ICE/DTLS
- Solution: piggyback SCTP init into the SDP exchange
 - base64 encoded SCTP INIT chunk
 - `a=sctp-init:AQAAHols3R0AUAAA/////B5ZR3AAAAEgAgABoLA`
 - Jump straight to DCEP or negotiated channels
- Implementations
 - libWebRTC/Chromium [tracking bug](#), behind flag
 - Pion [SCTP](#) / [WebRTC](#), waiting before merging