API

• Just like thewebsocketprotocol API
  – Send and receive data
  – Properties: protocol, binaryType, bufferedAmount

• Unidirectional SCTP streams are paired to form bidirectional data channels

• Per-message SCTP characteristics are set on the channel:
  – Reliability, ordering, payload protocol ID
How to use SCTP

• Classic SCTP usage patterns don’t negotiate per-message characteristics
• Application usage determines requirements for reliability, ordering, etc…
  – e.g., Diameter, M3UA, many 3GPP protocols
• Binding of request/response occurs in the application protocol
  – Common solution: transaction IDs
• Streams are used as needed
Negotiation Requirements

• Need:
  – Both peers agree to establish an SCTP association
  – Peers pair streams consistently

• Optional:
  – The same per-message characteristics in a channel
  – The same values for non-functional parameters
No In-Band Protocol

• In-band negotiation adds unnecessary overhead for all well-defined usage patterns or any with out-of-band negotiation
• draft-jesup-rtcweb-data-protocol-04 defines an open message that carries properties ahead of any data
  – Odd/even glare resolution
• draft-marcon-rtcweb-data-channel-management-00 defines a more generic configuration declaration
3. SDP-Only Option

m=application 12345 SCTP/DTLS 0 1 2 5
c=IN IP6 ::1
a=fmtp:0 binaryPPID=177; label=control
a=fmtp:1 label=chat
a=fmtp:2 label=chars; rtx=2; protocol=lrudfb
a=fmtp:5 label=bullets; rtxTime=5000
3. Offer/Answer

- Channel parameters are declarative
- Same-number streams form channels
- Peers are required to create a channel with the identified parameters
- Peers might have mismatched parameters if they already have a channel
  - Only happens if they don’t negotiate
3A. No Negotiation Option

• Zero RTT
• Just use channels
• Peers create channels when they first see data
• Select new stream numbers in a predicable fashion, i.e., lowest available number
• Require that channel parameters are mutable
QUESTIONS