RTP Usage for CLUE

IETF 82 – 14 November 2011

Jonathan Lennox jonathan@vidyo.com

Allyn Romanow allyn@cisco.com

Paul Witty pauwitty@cisco.com

Source Multiplexing: Motivation

- A telepresence session has lots of sources
  - Dozens at a time
    - e.g. for a continuous presence screen
  - Out of a pool of hundreds possible
- Sessions have asymmetric numbers of sources
- So the usual SIP model (a single media source per session) doesn't scale, and is needlessly complex.

Source multiplexing

- Send all sources (of the same media type) over a single RTP session, single transport flow.
  - Protocol behavior is straightforward
  - NAT traversal is fast, port consumption is low.
  - SDP is small, and looks “normal” to middleboxes.
- This was always part of RFC 3550 (RTP), but not widely used until recently.

Source multiplexing: exceptions

- There may be cases where we still need to use multiple RTP sessions
  - Most obviously, if sources should have different QoS.
  - This doesn't preclude having those sessions themselves carry multiple sources!

Source multiplexing: complications

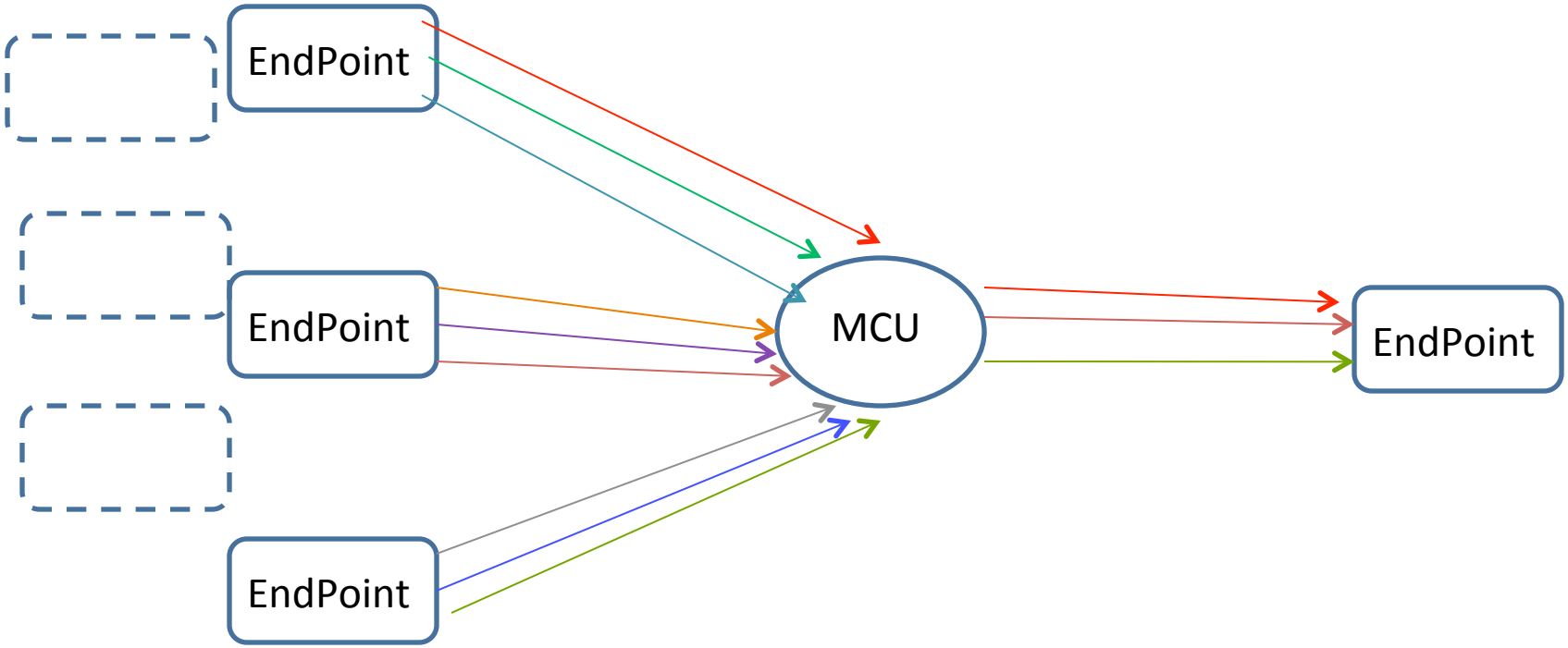
- Some things get complicated
  - One-source-per-session was a fairly pervasive assumption.
    - Even though RTP always supported source multiplex.
  - Details of RTCP behavior.
  - Backward compatibility.
- Not in scope for CLUE – general IETF architecture.
- See:
  - `draft-westerlund-avtcore-multiplex-architecture-00`
  - `draft-lennox-rtcweb-rtp-media-type-mux-00`
  - The AVTCORE WG, and probably other groups too.

CLUE-specific Complications

- When you receive a source, you need to know *why* you're receiving it.
  - Which requested capture it corresponds to.
- This can change dynamically
  - Source collision / restart.
  - Switched captures.
  - Source moving between switched captures.
    - Three camera to two screen: LC → LR → CR → LR
- This is needed before stream decoding starts.
  - Many systems: which screen to display on → which decoder hardware to use.

Demultiplexing RTP Streams in Same Session

Case to Consider: Infinite Sources



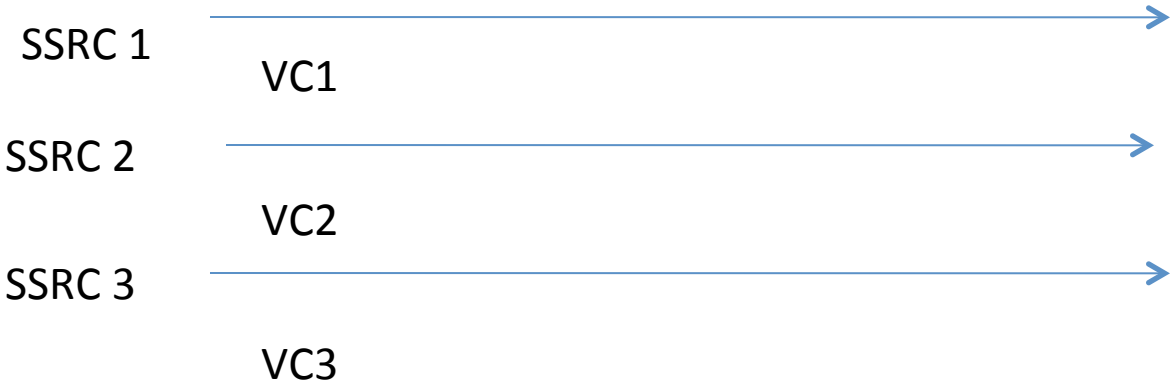
How to Demultiplex

- SSRC
- MuxID
- Hybrid

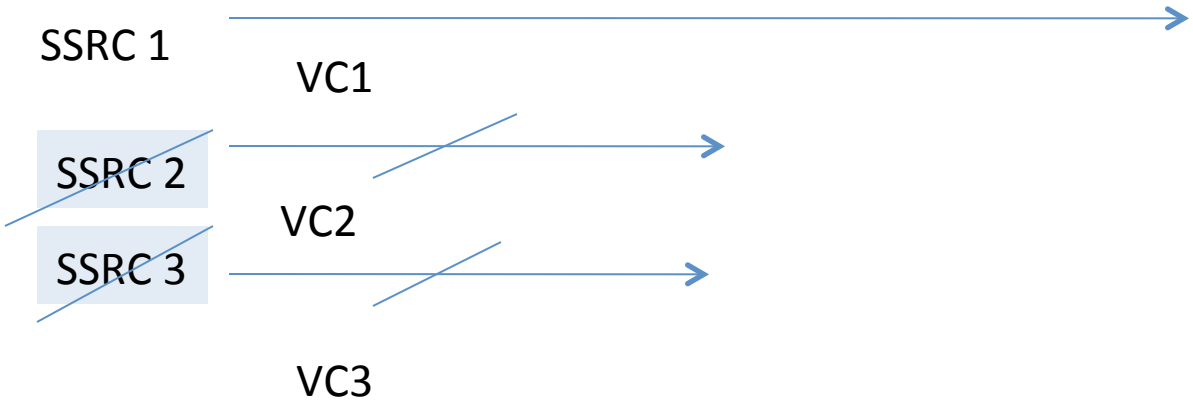
SSRCs Pros

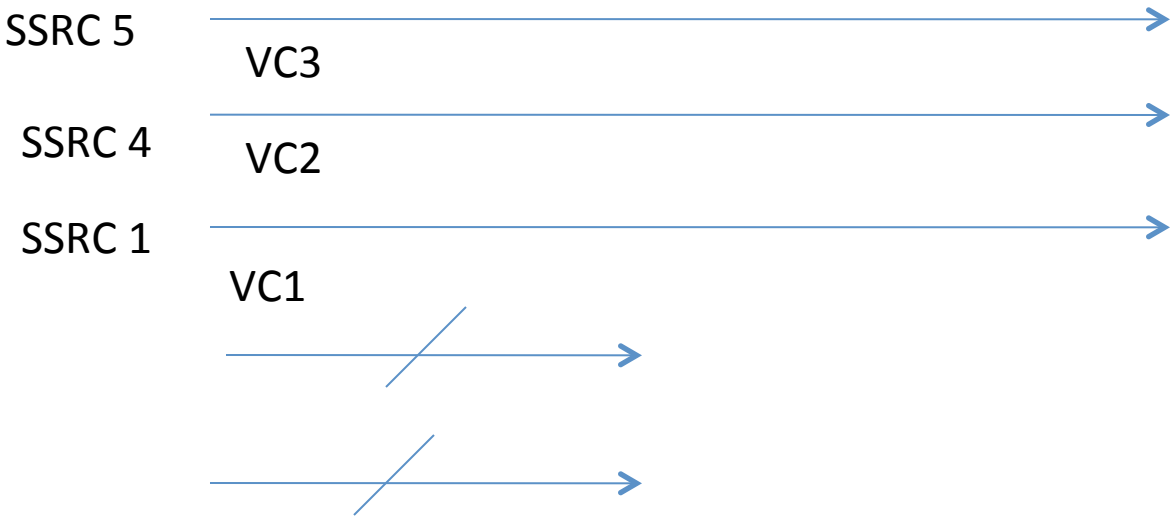
- Already in RTP packet
- Unique number

The issue with SSRC: 3 streams sent

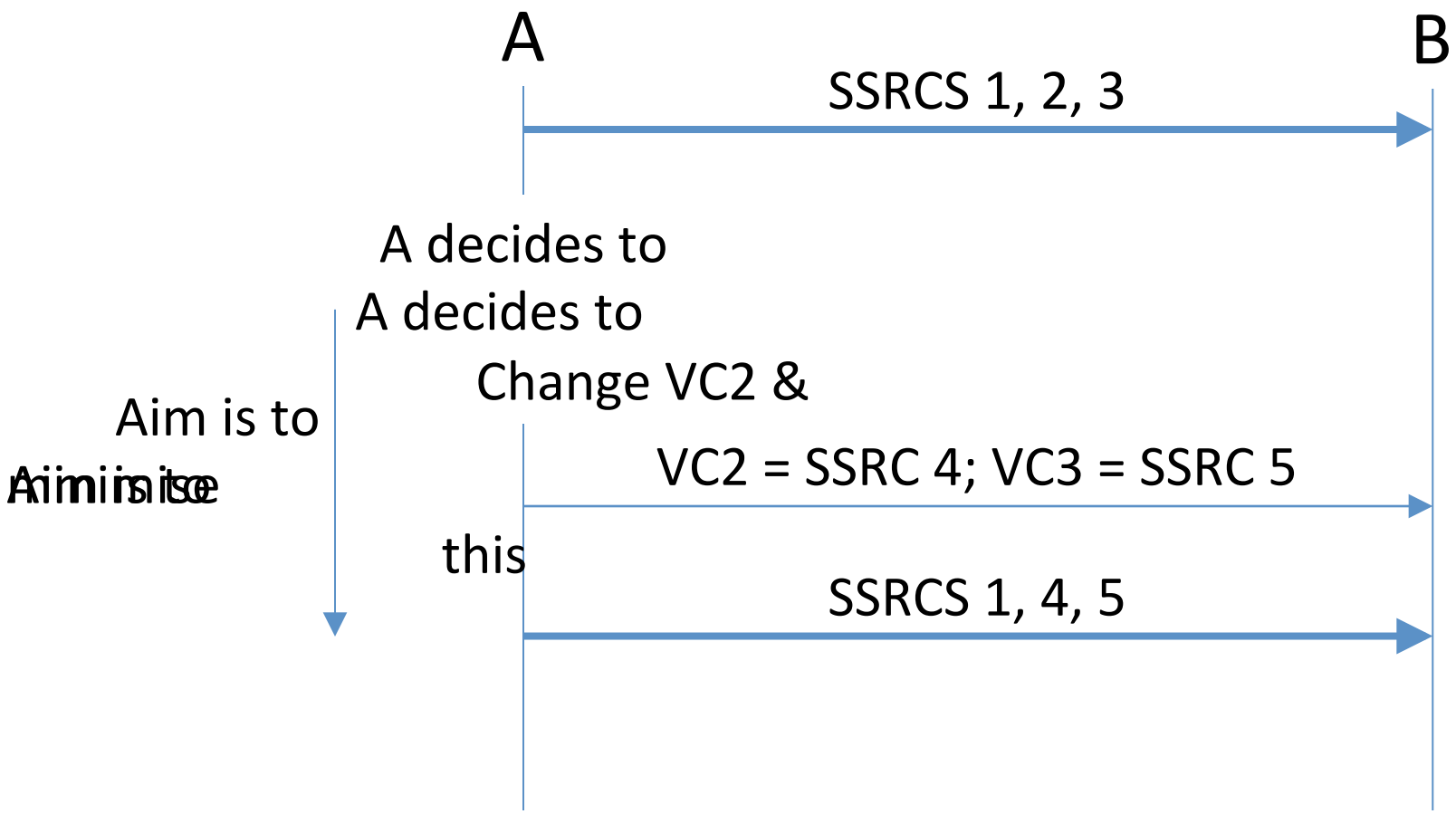


2 of 3 streams stop




SSRC 5
VC3
SSRC 4
VC2
SSRC 1
VC1

Timeline


A
B
SSRCS 1, 2, 3
A decides to
A decides to
Change VC2 &
Aim is to
VC2 = SSRC 4; VC3 = SSRC 5
this
SSRCS 1, 4, 5

SSRCs Challenges

- Timing requirements for codec when SSRC changes
- Timing requirements for codec when SSRC changes

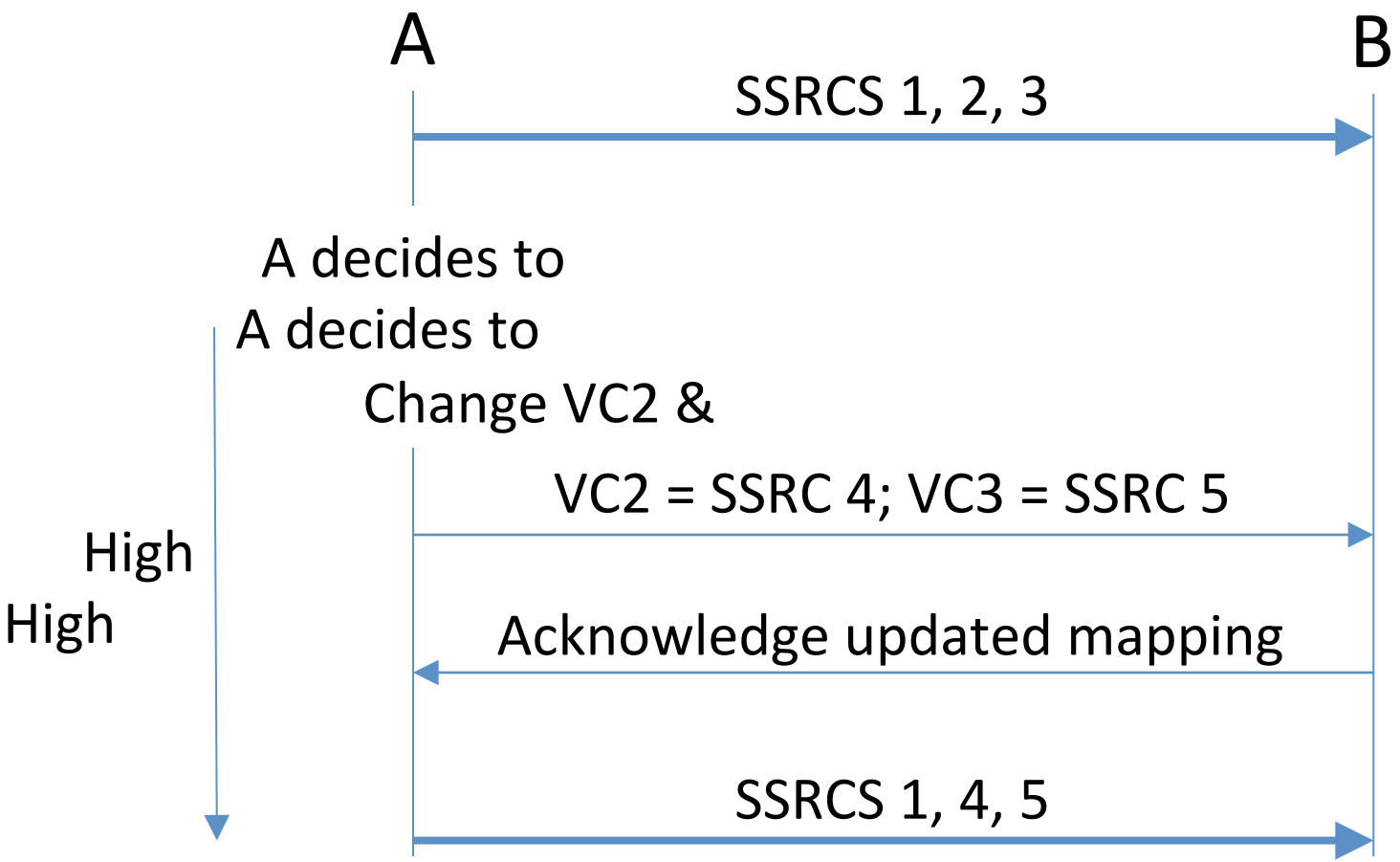
Where to put metadata

- Advantage – reliable
- Disadvantage – reliability can cause large delays, esp. when lots of conference participants. Switching latency

- RTCP, e. g., new SDES

  to route media
- Can use acks and retrans, but can cause high latency

Sending metadata



Multiplex ID Advantages

- Tag media packet with

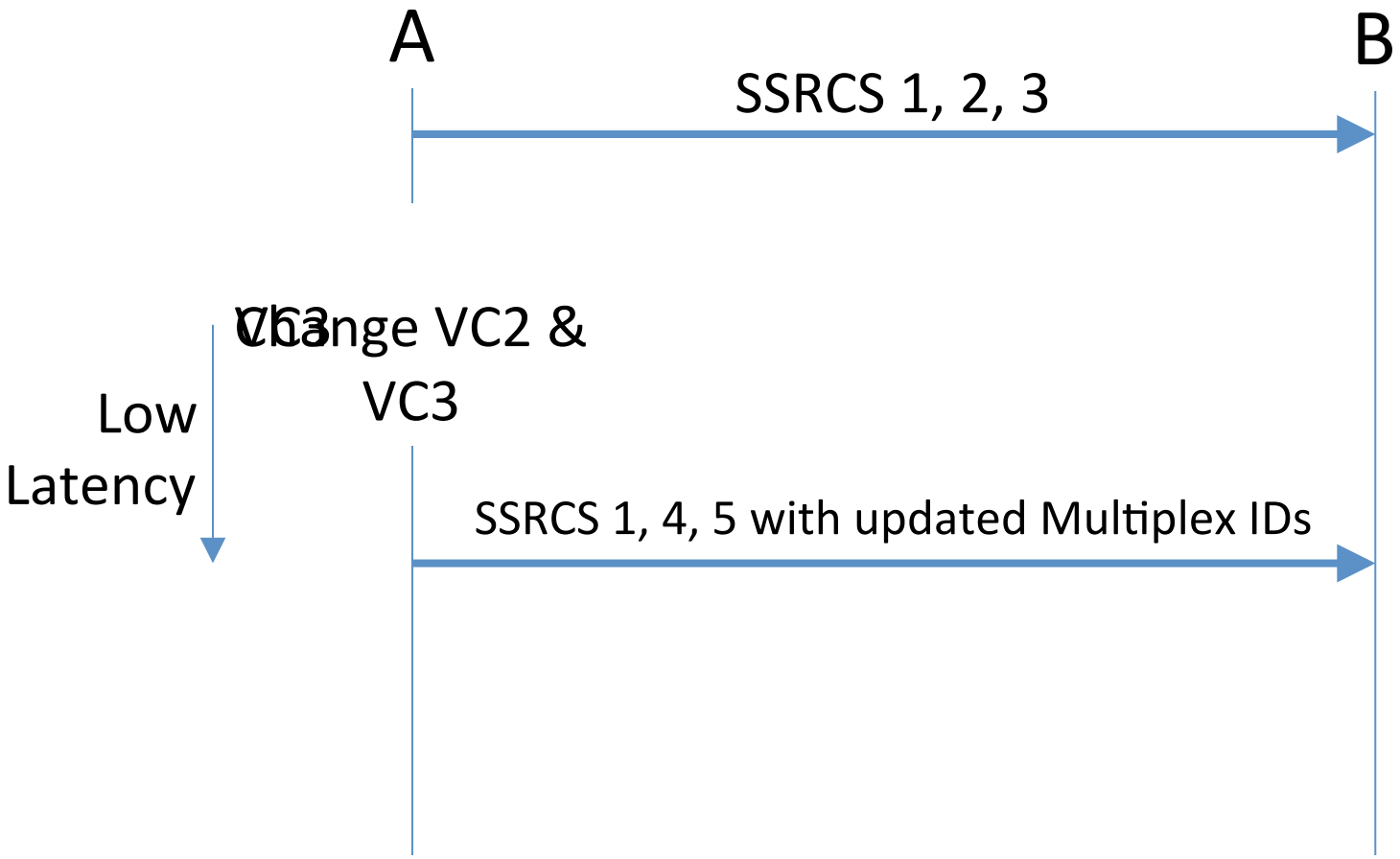
– Header extension

  - When stream changes, muxID
  - Advantage – no lag time when SSRC changes
  - Receiver can add useful info to a muxID

Multiplex ID Cons

- Disadvantage – high processing costs
  - Scoped only within one hop
  - Adding, modifying expensive due to SRTP auth
  - requires re-auth of whole packet, could limit throughput
  - Might need to re-auth due to SDP anyway

Sending Multiplex IDs



Hybrid Scheme, Pros and Cons

- MuxID

media (typically an IDR/GDR).

- All other packets demux

using only the SSRC

- Advantage - Mitigates high switching latency and high processing cost

Hybrid Challenges

- Needs more investigation