Generic RTP Multiplexing (GeRM)

Mark Handley
USC/ISI
mjh@isi.edu

Goals

- To be able to multiplex generic RTP streams

- When gateways cooperate, to have an overhead as low as one byte per multiplexed payload.

- When gateways don't cooperate, overhead depends on which traffic gets multiplexed together.
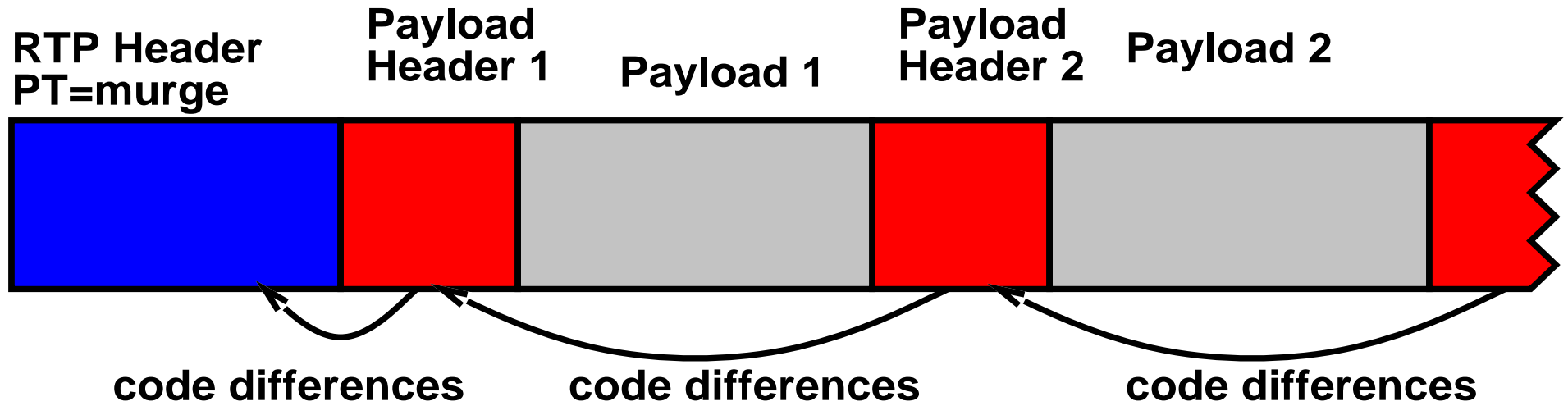
- Worst case: no worse than full RTP header per payload.

Status

- **First idea during Chicago IETF.**

- **Internet draft now available**
  - name changed from MuRGE to GeRM.

- **No changes required from Chicago slides.**
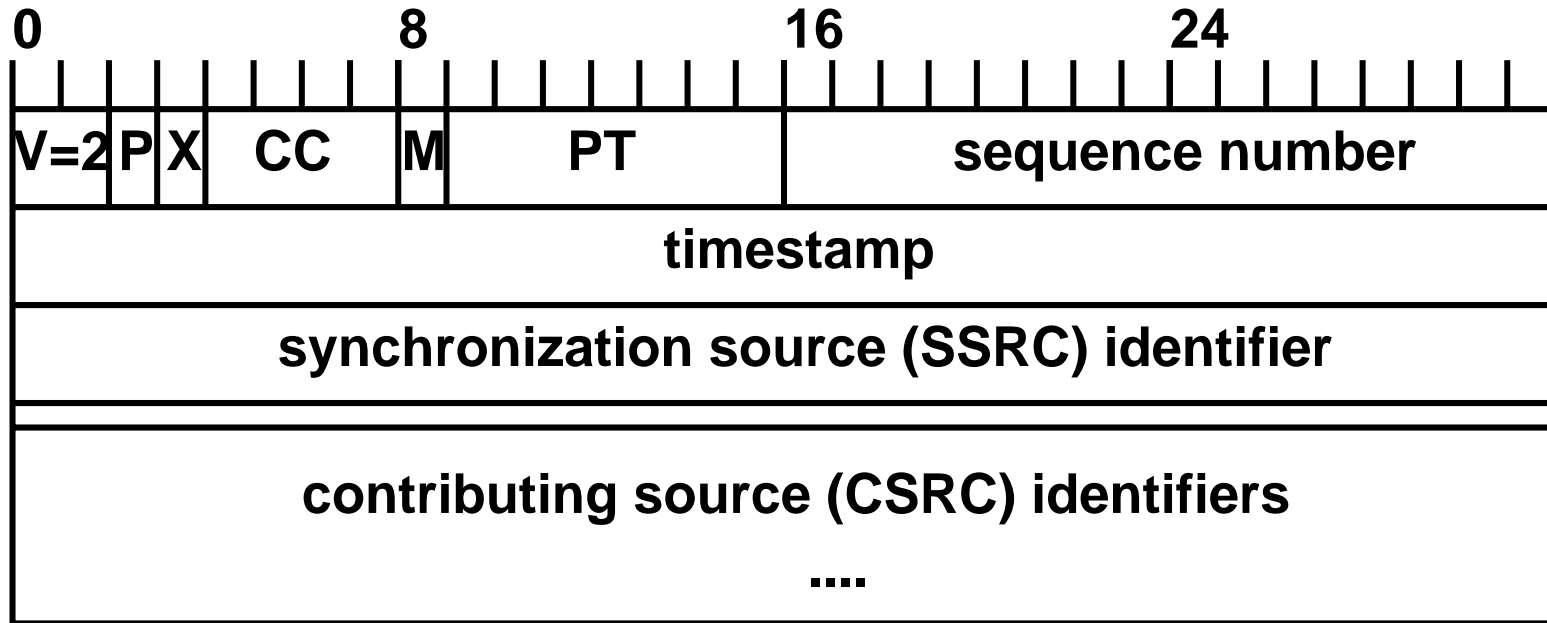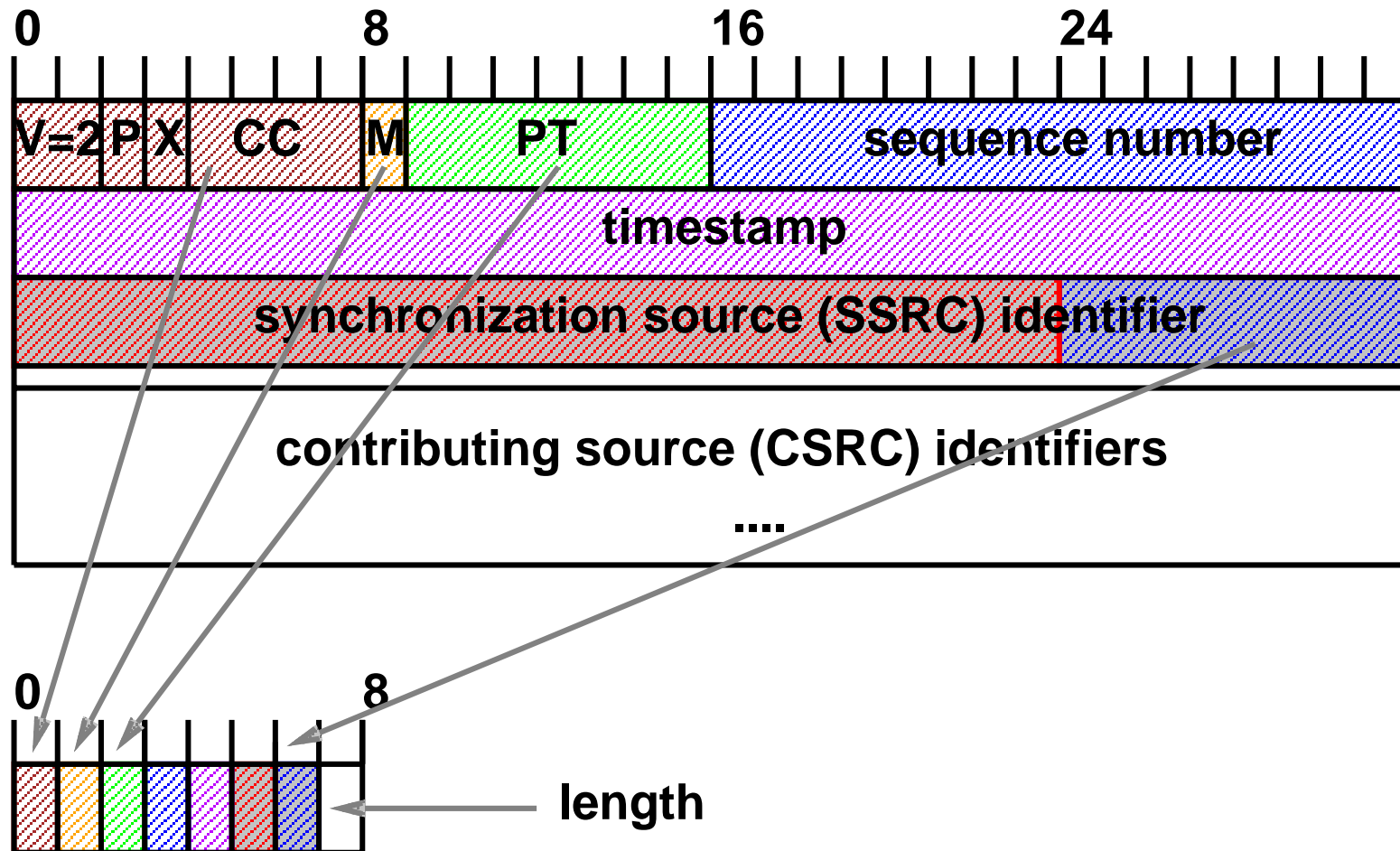 - Just improved the explanation and corrected typos.

GeRM Difference Coding

RTP Header
PT=murge

Payload
Header 1

Payload 1

Payload
Header 2

Payload 2

code differences   code differences   code differences

RTP Header

0				8		16	24	
V=2	P	X	CC	M	PT		sequence number	
timestamp								
synchronization source (SSRC) identifier								
contributing source (CSRC) identifiers								

GeRM Header

GeRM Bits

- **Bit 0:**
  - ▸ zero=>Byte 1 unchanged (V=same, P=zero, CC=same)
  - ▸ one=>Byte 1 follows.
- **Bit 1:**
  - ▸ zero=>PT unchanged, one=>PT follows
- **Bit 2:   M bit**
- **Bit 3:   zero=>seq number unchanged**
- **Bit 4:   zero=>timestamp unchanged**
- **Bit 5:   zero=>SSRC (3 MS bytes unchanged)**
- **Bit 6:   zero=>SSRC (LS byte increased by one)**
- **Bit 7:**
  - ▸ zero=>length unchanged
  - ▸ one=>one byte length field follows

Usage

- **GeRM can be used between gateways with no additional signalling.**
  - ▸ Approx 11 bytes per payload, vs 40 for IP/UDP/RTP.
- **GeRM can be used with an additional signalling protocol that performs SSRC/SeqNo/Timestamp mappings.**
  - ▸ Can reduce overhead to 1 byte per payload in the limiting case of same PT, same length payloads, no holes in the (mapped) SSRC space.
  - ▸ Normal case for POTS->IPTEL->POTS would be approx 7 bytes if mapping is only done at call startup.

GeRM signalling

If we decide that this approach is promising, we should probably also specify a gateway->gateway signalling protocol.

- **Allow robust SSRC mapping at call startup.**
- **Allow robust SSRC remapping when call terminates and leaves a space in the SSRC space.**
  - May not be required if call startup rate is high.
- **Allow seqno remapping?**
  - More difficult - issue is one of silence supression.
- **Allow timestamp remapping due to clock drift?**
  - Only an issue for RTP not generated at the gateway.

What Are Our Goals?

- **If we want a single mux protocol, probably we want something like GeRM.**

- **If we really care about a couple of bytes per payload, GeRM may not be efficient enough.**
  - ▸ Probably have to have several different special purpose multiplexing protocols.

- **I'm not going to push GeRM forward without concensus from this group that a general RTP multiplexing protocol is needed.**
  - ▸ What's the concensus?