UDP CS rules

• **ATOMIC**
  – UDP CS if acting as L4
  – OMIT if acting as L2

• **Non-ATOMIC**
  – Never UDP CS each fragment
  – Reassembled result
    • Reassembled CS required if this is the “real” L4
    • OMIT if this layer acts as L2
Reasoning – in order

1. IPv6 assumes L2/L4 checks for errors

2. Tunnels are L2s - but over other L2s

3. Nodes don’t mangle, transmission and reassembly does
   i.e., true L2s check transmission; all L2s check reassembly
   so tunnels that don’t reassemble don’t need checks

4. UDP CS needed when L4;
   UDP CS needed when L2
   only upon reassembly and only on that result

Some details follow…
1. IPv6 assumes others check

• IPv6 has header-only checksum
  – Assumes L2 checks hop errors
  – Assumes L4 checks E2E errors
  – Because “nodes don’t mangle,”
    but packets (at any layer) can be lost

• Which is why UDPv6 requires UDP CS
  – UDPv6 refers to UDP in the IPv6
  – UDPv6 is thus L4
  – L4 needs to check E2E errors

• And why UDP as L2 requires UDP CS
  – But only when there isn’t another layer that does so already…
2. A tunnel is L2

• They look the same:
View from inside the next layer

• Link or tunnel
  – both look like a host-to-host path at layer N
  – ingress/egress or interfaces look like hosts
View from the upper layer

• Both look like a (L2) hop in a (L3) network $M$
3. Reassembly vs. checks

• Transmission as reassembly
  – Packets become sequence of symbols
  – Symbols represent bits or groups of bits
  – Transmission is reassembly of these groups

• Other reassembly
  – None for “atomic” packets (not fragmented)
  – IPv6 reassembles (but doesn’t check)
    • Still relies on L2 to check chunks, L4 to check whole
4. UDP CS rules

• **ATOMIC**
  - UDP CS if acting as L4
  - OMIT if acting as L2

• **Non-ATOMIC**
  - Never UDP CS each fragment (*wasted effort!*)
  - Reassembled result
    - Reassembled CS required if this is the “real” L4
    - OMIT if this layer acts as L2

Current IPv6 rules