IPTFS Reorder/lost frame issue

Tero Kivinen <kivinen@iki.fi>

Section 2.5 of IPTFS draft:

2.5. Summary of Receiver Processing

An AGGFRAG enabled SA receiver has a few tasks to perform.

The receiver first reorders, possibly out-of-order ESP packets received on an SA into in-sequence-order AGGFRAG_PAYLOAD payloads (Section 2.2.3). If congestion control is enabled, the receiver considers a packet lost when it's sequence number is abandoned (e.g., pushed out of the re-ordering window, or timed-out) by the reordering algorithm. As an optional optimization (e.g., to handle very lossy and/or reordered tunnel paths), the receiver MAY transmit any fully formed inner packets contained within the AGGFRAG_PAYLOADs prior to re-ordering the outer packets.

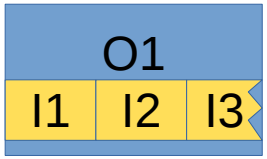
Additionally, if congestion control is enabled, the receiver sends congestion control data (Section 6.1.2) back to the sender as described in Section 2.4.2 and Section 3.

Finally, the receiver processes the now in-order AGGFRAG_PAYLOAD payload stream to extract the inner-packets (Section 2.2.3, Section 6.1).

Issues in section 2.5

- It implies that normal processing is to reorder outer ESP packets to in-order stream and process them after that.
- New text was added in -11 version to allow optimization where receiver MAY transmit any fully formed inner packets before re-ordering.
- I think this optimization should be default, and the in-order processing should not be used in normal cases.

Normal flow



Receiver receives O1, starts processing it.

Outer packet

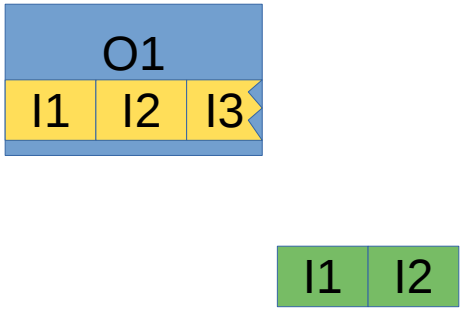
Complete inner packet

Start of the inner packet

End of the inner packet

Inner packet out from the device

Normal flow



Sends out inner packets I1 and I2, cannot send I3, as it is not fully received

Ox — Outer packet

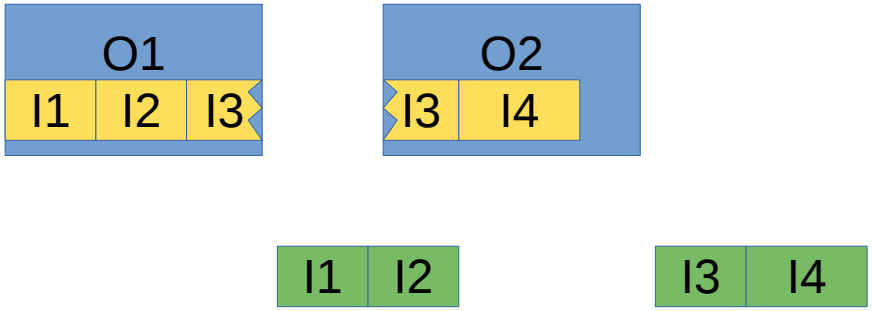
Ix — Complete inner packet

Ix — Start of the inner packet

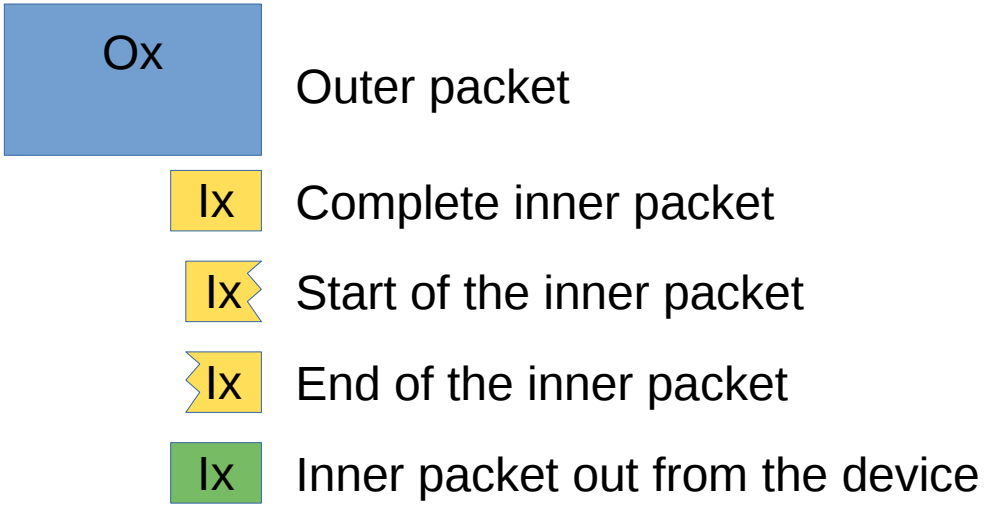
Ix — End of the inner packet

Ix — Inner packet out from the device

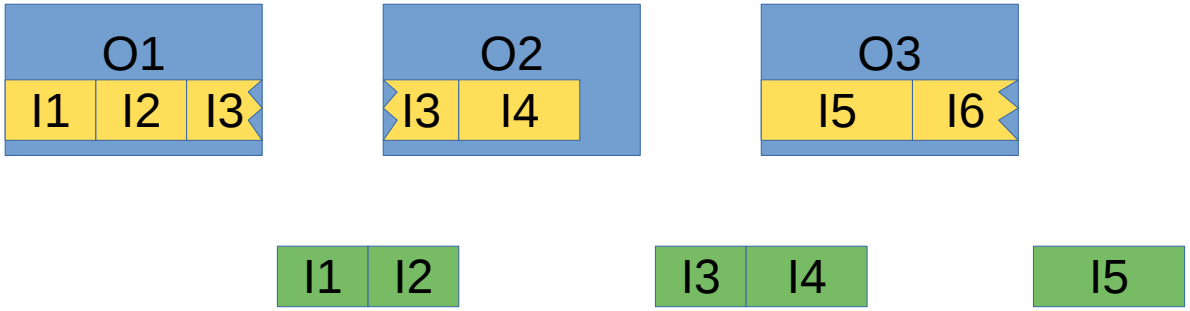
Normal flow



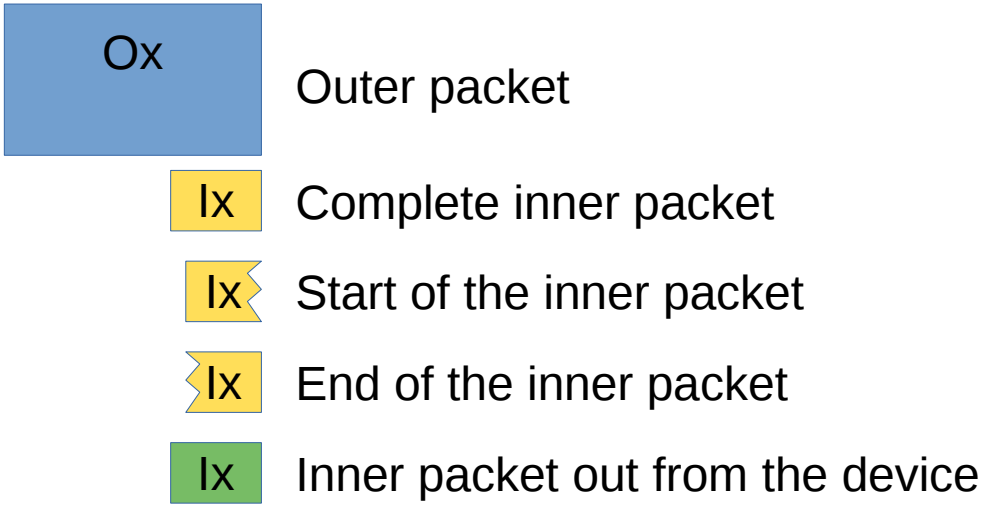
Receiver receives O2, sends out nowcomplete I3, and I4



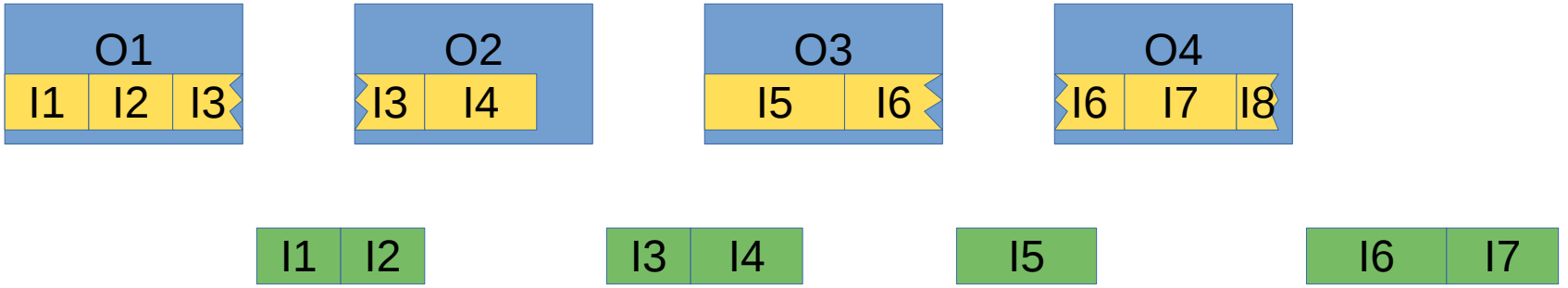
Normal flow



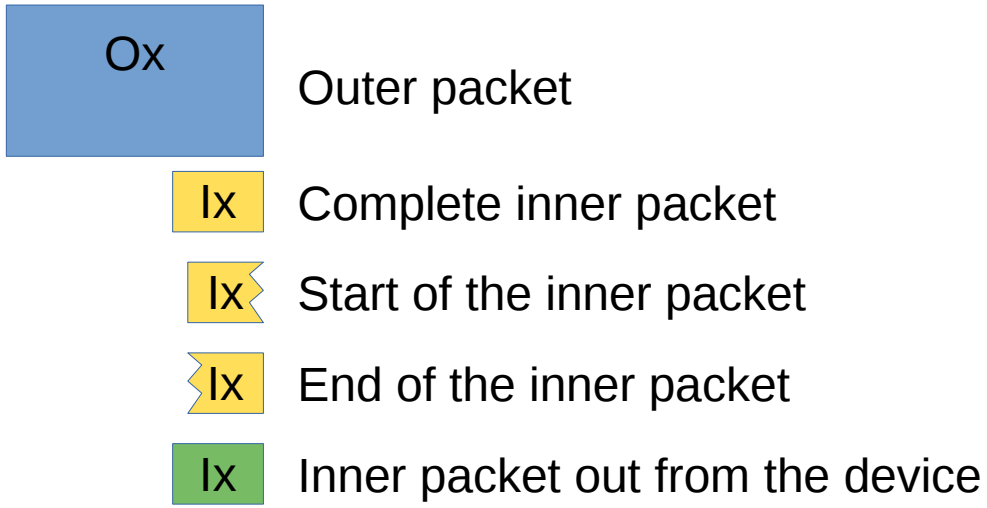
Receiver receives O3, Sends out I5



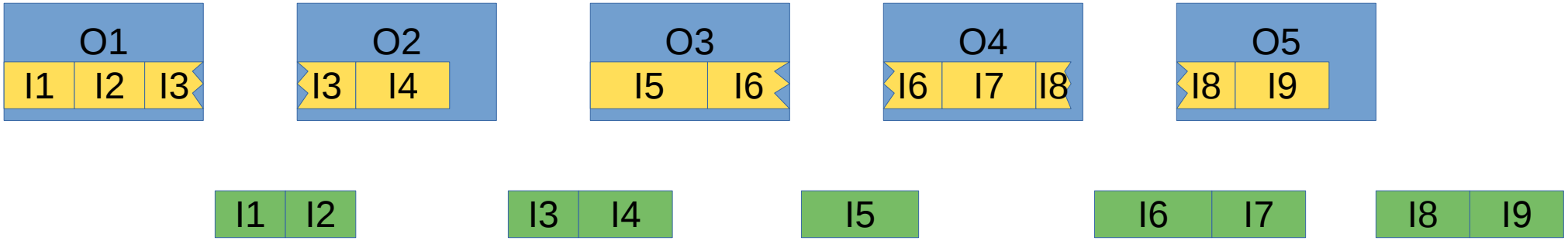
Normal flow



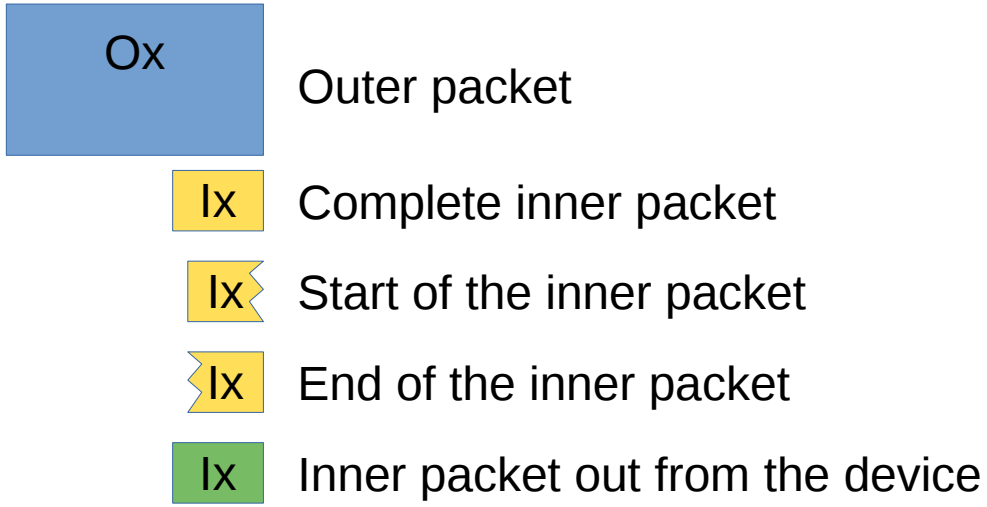
Receiver receives O4, Sends out I6, and I7



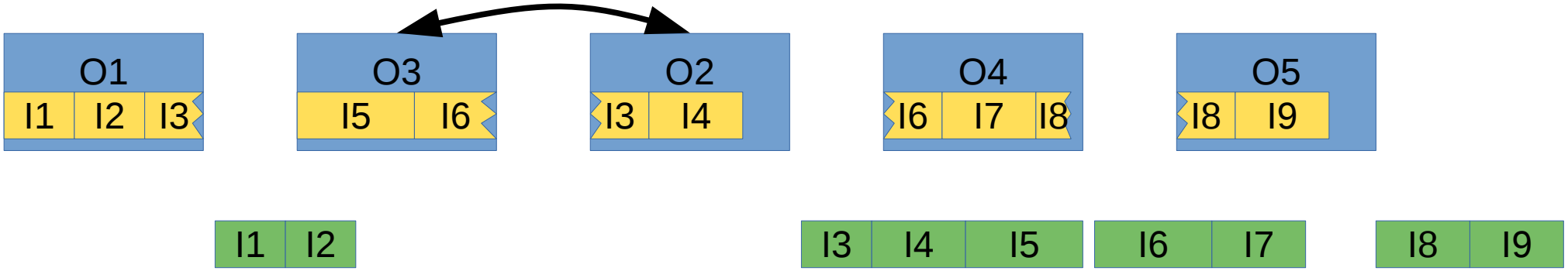
Normal flow



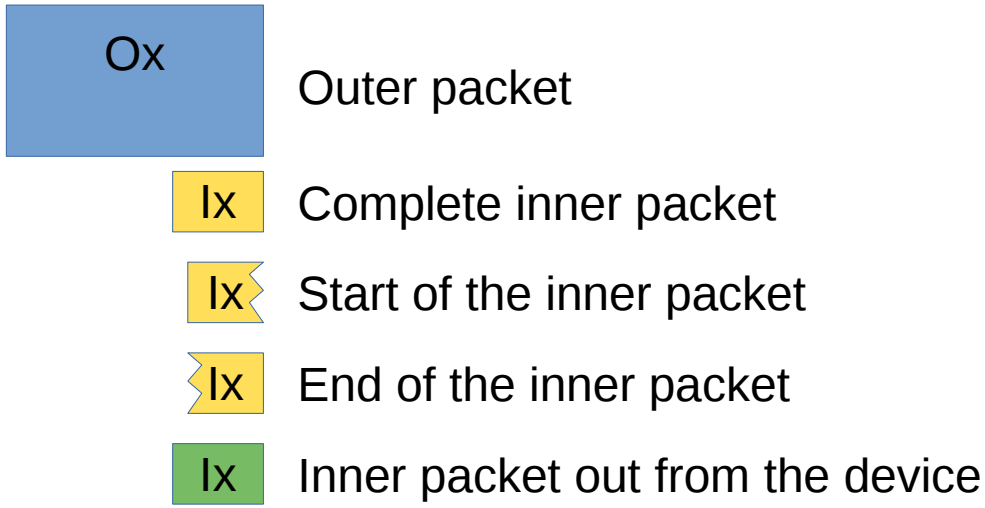
Receiver receives O5, Sends out I8, and I9



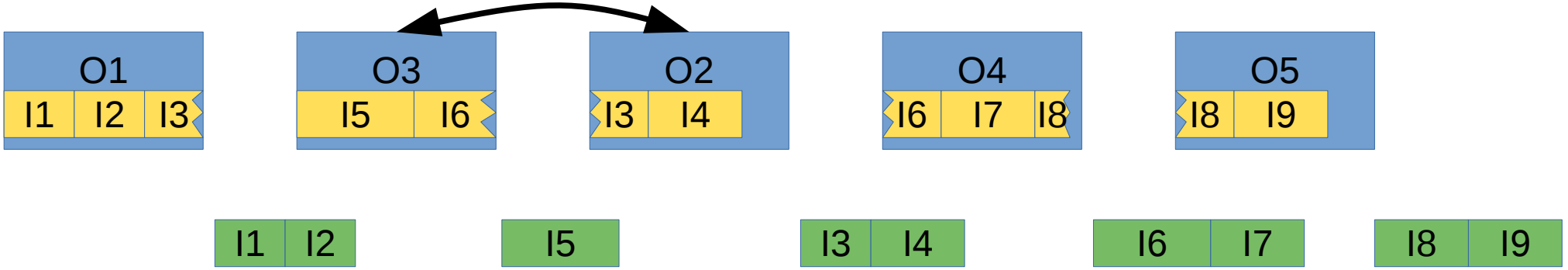
Reordered flow (in-order)



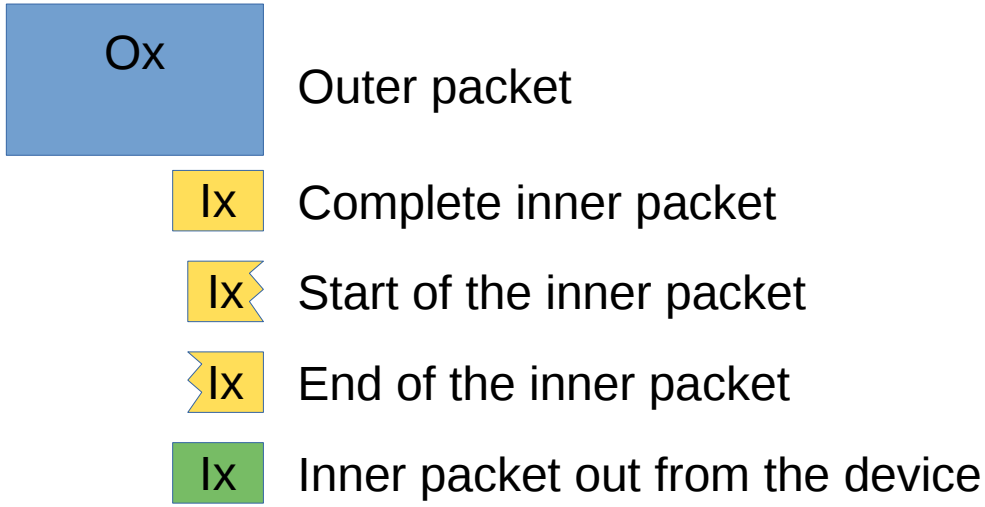
When receiver receives O3, it buffers it. When it receives O2, it processes O2, and then O3, thus fully complete I5 will be delayed until O2 arrives to keep order.



Reordered flow (immediate)

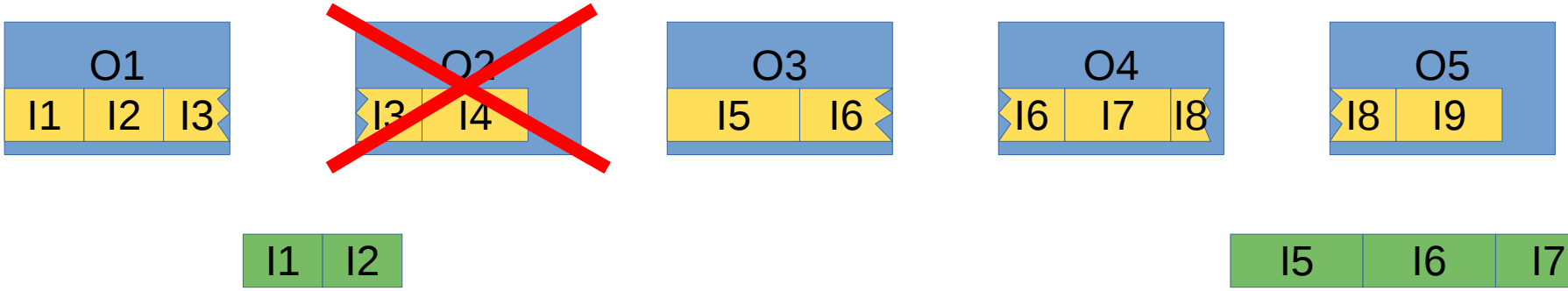


When receiver receives O3, it buffers I6, but sends I5 out as it is complete. When it receives O2, it reassembles I3, and sends out, and then sends I4. I5 is sent before I3 and I4, meaning the reordering in outer frames is kept and is visible in inner frames too.

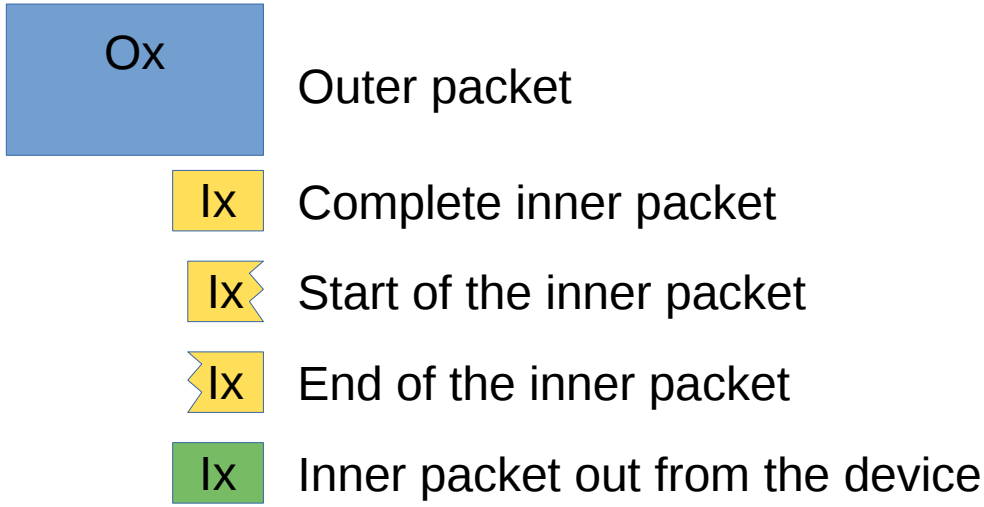


Lost frame (in-order)

Assuming reorder window of 2

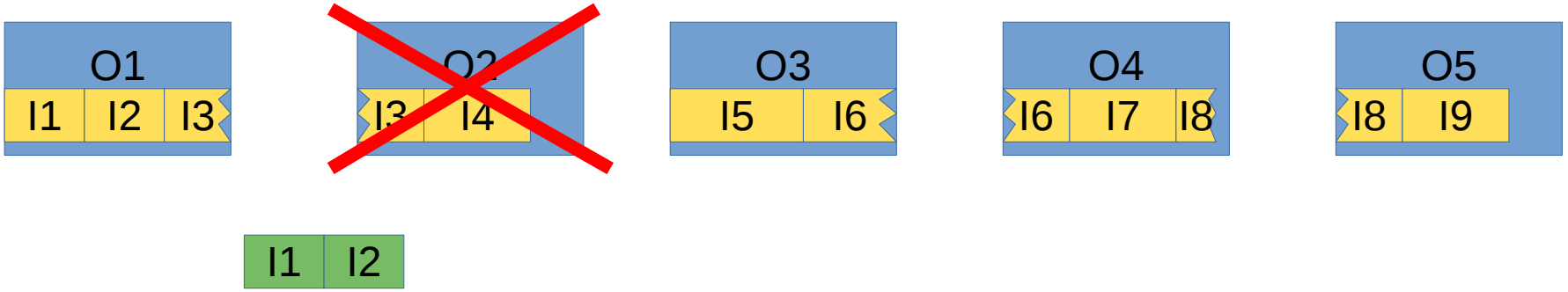


When receiver receives O3, it realises frame is missing, so it buffers O3 until reorder buffer gets full. If reorder window is 2, after receiving O4 it knows O2 was lost, and only after that it will send out I5, I6, and I7. Receiver needs to buffer I3 of O1, and O3 until they fall out from reorder window, thus reorder window greatly affects memory usage.

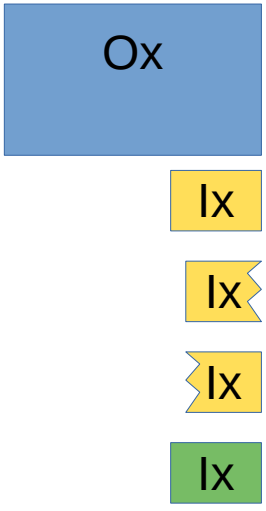


Lost frame (in-order)

Assuming reorder window of 3

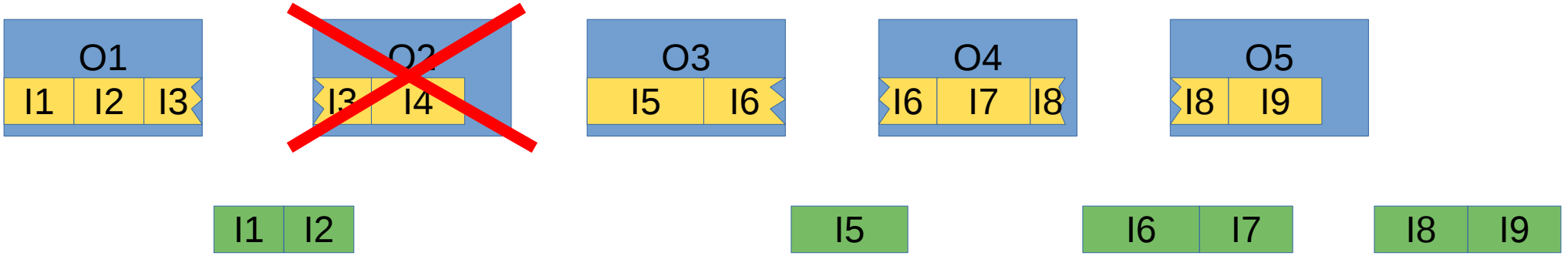


When receiver receives O3, it realises frame is missing, so it buffers O3 and O4 until reorder buffer gets full. If reorder window is 3, after receiving O5 it knows O2 was lost, and only after that it will send out I5, I6, I7, I8, and I9. Receiver had to buffer I3, O3, and O4. Larger the reorder window greater the delay and memory usage.



Lost frame (immediate)

Reorder window does not affect processing



When receiver receives O3, it processes it normally and sends out I5. It processes O4, and O5 normally and send the inner packets out when they are complete. No buffering happening. Large reorder window does not affect delay, and only I3 from O1 is buffered until O2 falls out from the reorder window and then it is discarded.