Requirements for Scalable Transport	**Normative text**	**SCReAM**	**Evaluation/Remarks/Plans/Issues/Objections**
1. Use of L4S Packet Identifier (A1.1)	Section 4.1: A sender that wishes a packet to receive L4S treatment as it is forwarded, MUST set the ECN field in the IP header (v4 or v6) to the ECT(1) codepoint.	Partially Compliant ECT(1) code point setting for SCReAM is straightforward.	RFC82998 is not L4S capable, the running code on https://github.com/EricssonResearch/scream is however L4S capable. An update of RFC8298 is pending more evaluation of SCReAM with/without L4S support. There are potential issues with setting/reading the ECN bits on non-Linux OS stacks. These are however implementation issues. SDP negotiation to set ECT 1 should follow RFC6679, this is not implemented in current running code. This is not necessary in experiment platforms but it is needed e.g. for the cases that SCReAM with L4S is implemented in media streaming platform such as WebRTC.
2. Accurate ECN Feedback (A1.2)	Section 4.2: For a transport protocol to provide scalable congestion control it MUST provide feedback of the extent of CE marking on the forward path.	Compliant SCReAM uses RFC8888 for the RTCP feedback. RFC echoes the ECN bits for each received RTP packet	
(Scalable CC requirement)	Section 4.3: As a condition for a host to send packets with the L4S identifier (ECT(1)), it SHOULD implement a congestion control behaviour that ensures that, in steady state, the average time from one ECN congestion signal to the next (the 'recovery time') does not increase as flow rate scales, all other factors being equal.	Compliant SCReAM in L4S mode implements backoff similar to DCTCP, i.e with an alpha factor that is updated with the fraction of CE marked packets every RTT.	
(ECT(1) use needs Prague compliance)	Section 4.3: In order to coexist safely with other Internet traffic, a scalable congestion control MUST NOT tag its packets with the ECT(1) codepoint unless it complies with the following bulleted requirements.		
(Prague compliance description)	Section 4.3: The specification of a particular scalable congestion control MUST describe in detail how it satisfies each requirement, and for any non-mandatory requirements, it MUST justify why it does not comply.		
3. Fall back to Reno-friendly congestion control on packet loss (A1.3)	Section 4.3: As well as responding to ECN markings, a scalable congestion control MUST react to packet loss in a way that will coexist safely with a TCP Reno congestion control [RFC5681].	Compliant SCReAM backs off on packet loss, but not with a factor 0.5. Rather it scales down the congestion window with a factor 0.8.	The rationale behind the reduced backoff to loss in SCReAM is that the encoded video has a variable frame size and that gives some additional headroom to avoid that the larger frames build up a large queue, the effect is that SCReAM is most often bordering to underutilizing link capacity. The outcome is therefore that SCReAM can coexist safely with Reno.
4. Fall back to Reno-friendly congestion control on classic ECN bottlenecks (A1.4)	Section 4.3: A scalable congestion control MUST implement monitoring in order to detect a likely non-L4S but ECN-capable AQM at the bottleneck. On detection of a likely ECN-capable bottleneck it SHOULD be capable (dependent on configuration) of automatically adapting its congestion response to coexist with TCP Reno congestion controls [RFC5681]. To participate in the L4S experiment, a scalable congestion control MUST be capable of being replaced by a Classic congestion control (by application and by administrative control).	Partially Compliant SCReAM use delay based congestion control. The estimated queue delay will be near zero when L4S bottlenecks are encountered. For cases with classic ECN queues in the network, the queue delay would be higher which means that it should be possible to detect non-L4S but ECN capable AQMs.	Methods for the detection and fallback to classic ECN are currently not implemented.

5. Reduce RTT dependence (A1.5)	Section 4.3: A scalable congestion control MUST eliminate RTT bias as much as possible in the range between the minimum likely RTT and typical RTTs expected in the intended deployment scenario.	Partially Compliant to Non-compliant New evaluations are needed, possibly with improved RTT bias mechanisms	This has drawn less focus and needs to be evaluated and possibly addressed. It is here likely that the algorithms devised for Prague can be used. RTT bias was evaluated in the RMCAT work (TC 5.5 in https://datatracker.ietf.org/meeting/96/materials/slides-96-rmcat-0) but new evaluations are likely needed.
6. Scaling down to fractional congestion windows (A1.6)	Section 4.3: A scalable congestion control SHOULD remain responsive to congestion when typical RTTs over the public Internet are significantly smaller because they are no longer inflated by queuing delay.	Partially Compliant to Non-compliant The minimum congestion window is 3 MSS (configurable).	SCReAM implements packet pacing but SCReAMs performance in very low RTT deployments is not yet evaluated.
7. Measuring Reordering Tolerance in Time Units (A1.7)	Section 4.3: A scalable congestion control SHOULD detect loss by counting in time-based units, which is scalable, as opposed to counting in units of packets (as in the 3 DupACK rule of RFC 5681 TCP), which is not scalable. This requirement does not apply to congestion controls that are solely used in controlled environments where the network introduces hardly any reordering.	Partially Compliant SCReAM implements loss detection in time based units (fixed 10ms) . The span of the RTCP feedback is however typically limited (64packets in current implementation), this limits the allowed reordering depth to avoid a spurious loss detection.	RFC8888 allows larger RTCP feedback spans than the 64 packets that is used in the running code, alternatively other RTCP feedback extensions may be used to avoid spurious loss detection due to large reordering
(Burst limit)	Section 4.3: A scalable congestion control is expected to limit the queue caused by bursts of packets. It is only required that the specification of a particular scalable congestion control MUST define, quantify and justify its approach to limiting bursts.	Partially Compliant SCReAM by default implements packet pacing..	Lately it has been experimented with microburst pacing wherein packets are transmitted in bursts with e.g. 2 or 5ms intervals. The reason to this is that it can help to reduce power consumption in 5G phones
Scalable Transport Protocol Optimizations	**Appendix text (no normative refs)**		
1. Setting ECT in TCP Control Packets and Retransmissions (A2.1)	This item only concerns TCP and its derivatives (e.g. SCTP), because the original specification of ECN for TCP precluded the use of ECN on control packets and retransmissions. To improve performance, scalable transport protocols ought to enable ECN at the IP layer in TCP control packets (SYN, SYN-ACK, pure ACKs, etc.) and in retransmitted packets. The same is true for derivatives of TCP, e.g. SCTP.	Unclear if this is relevant for RTP/UDP with RTCP as feedback mechanism RTP retransmissions are marked ECT(1), if L4S enabled RTCP is likely not ECT(1)	
2. Faster than Additive Increase (A2.2)	It would improve performance if scalable congestion controls did not limit their congestion window increase to the standard additive increase of 1 SMSS per round trip [RFC5681] during congestion avoidance. The same is true for derivatives of TCP congestion control, including similar approaches used for real-time media.	Partially Compliant SCReAM implements a fast increase mode that is entered if congestion is not experienced within one second..	The fast increase feature is described in RFC8298 but is being continuously evaluated and is subject to experimentation. Furthermore the rampup speed in the media rate control part in SCReAM is being evaluated as L4S gives a possibility for faster rate increase.
3. Faster Convergence at Flow Start (A2.3)	Particularly when a flow starts, scalable congestion controls need to converge (reach their steady-state share of the capacity) at least as fast as Classic congestion controls and preferably faster. This affects the flow start behaviour of any L4S congestion control derived from a Classic transport that uses TCP slow start, including those for real-time media.	Partially Compliant Test activities with the SCReAM BW test tool show that SCReAM with L4S can do flow start faster as the L4S marking helps to avoid large overshoot.	More evaluations with real video is however needed to verify that other side effects don't emerge.