Requirements for Scalable Transport	Normative text	Nvidia Geforce Now Compliant / Partially Compliant / Non-compliant	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.	Compliant Need OS/Kernel support.	Support in all platforms is not common, requires design changes to accommodate all platforms.
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	We need more clarification about timing of the feedback and RTT requirments
(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	The requirement is not clear. We like more clarification on RTT time, recovery time requirements and operating bitrate. The response time may vary based on multiple factors.
(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.	Compliant	See
(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.	Non-compliant	Is this requirement really needed? We may not be able to disclose implementation details on proprietary design.
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].	Partially Compliant	Although it is generally compliant, however our implementation tries to identify packet loss patterns and it may not react to all type of packet loss. Marking as partially-complaint.
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).	Compliant A monitoring scheme can be implemented based on OWD (one way delay)/RTT (Round trip delay) fluctuation heuristics. A fallback mechanism needs to be implemented.	Too many false hits. Needs real deployment experience to understand the real extend of this issue, and to tune/customize the settings based on dedicated experienced problems. Safety issues should be handled more on an operational level (A/B testing, active probing, network monitoring, …).
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.	Compliant Some of our use cases require streaming delay sensitive content on wireless networks which have high latency variation on airlink. We have implemented methods to monitor timing fluctuation as accurately as possible, but there the measurements will be noisy.	In networks with high jitter, this may not work as expected. Also for application with streaming over networks with multiple access points (i.e. Enterprise wifi/cellular, etc) the RTT bias may frequently change.
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.	Compliant	Our congestion control constantly monitoring channel condition and is complying. Note that we use UDP for streaming application.

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.	Compliant	
(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.	Compliant	Need a bit more clarification on pacing rate and time gap between group of packets. Also definition of bursts can be helpful (i.e. as how many packets).
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.	Compliant	Nvidia uses UDP for delay sensitive content.
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.	Compliant This is not relevant to our use cases.	Nvidia uses UDP for delay sensitive content.
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.	Compliant This is not relevant to our use cases.	Nvidia uses UDP for delay sensitive content.