Thin-streams and TCP

A wide range of Internet-based services that use reliable transport protocols display what we call thin-stream properties. This means that the application sends data with such a low rate that the retransmission mechanisms of the transport protocol are not fully effective. In time-dependent scenarios (like online games, control systems, stock trading etc.) where the user experience depends on the data delivery latency, packet loss can be devastating for the service quality. Extreme latencies are caused by TCP's dependency on the arrival of new data from the application to trigger retransmissions effectively through fast retransmit instead of waiting for long timeouts.

After analysing a large number of time-dependent interactive applications, we have seen that they often produce thin streams and also stay with this traffic pattern throughout its entire lifespan. The combination of time-dependency and the fact that the streams provoke high latencies when using TCP is unfortunate.

In order to reduce application-layer latency when packets are lost, a set of mechanisms has been made, which address these latency issues for thin streams. In short, if the kernel detects a thin stream, the retransmission mechanisms are modified in the following manner:

1. If the stream is thin, fast retransmit on the first dupACK.
2. If the stream is thin, do not apply exponential backoff.

These enhancements are applied only if the stream is detected as thin. This is accomplished by defining a threshold for the number of packets in flight. If there are less than 4 packets in flight, fast retransmissions can not be triggered, and the stream is prone to experience high retransmission latencies.

Since these mechanisms are targeted at time-dependent applications, they must be specifically activated by the application using the TCP_THIN_LINEAR_TIMEOUTS and TCP_THIN_DUPACK IOCTLS or the tcp_thin_linear_timeouts and tcp_thin_dupack sysctls. Both modifications are turned off by default.

References

More information on the modifications, as well as a wide range of experimental data can be found here:

"Improving latency for interactive, thin-stream applications over reliable transport"
http://simula.no/research/nd/publications/Simula.nd.477/simula_pdf_file