Ossification: a result of not even trying?

Michael Welzl, Gorry Fairhurst, David Ros

The Internet transport layer's lack of end-to-end'ness has inspired people to try to work against the influence of certain middleboxes. It has become common to use HTTP as a substrate [RFC 3205]. There have also been proposals to realize transport services that up to now could not be provided by TCP, some have built new transports on top of "UDP" (e.g. QUIC) and some have proposed something that looks as much as possible like "normal TCP" on the wire (MPTCP [RFC6824], Minion [1], "Inner Space for TCP Options", which puts options in the TCP payload [2]).

These methods may be workable solutions; however, they are technical compromises (header overhead, functional limitations, sometimes incomplete or unstable specifications, etc.). We therefore argue that it is important that such proposals are only used as a fallback, and not as the default solution.

Two ways in which middleboxes can get in the way of transports are: functions to enhance security, and functions that interact with specific protocols. By their nature, packets that constitute attacks can be "unknown" (not expected). With a security-driven way of configuring devices, it therefore becomes a natural decision to essentially block everything that is unknown. Such behavior is at odds with the "be liberal in what you accept" principle, and has led to trouble in the past (e.g. with PMTUD [RFC 1191, RFC 4821]). However, merely lamenting that this behavior is wrong does not solve anything – e.g. it is common for firewall software to only allow "known things" by default.

Blocking "everything that is unknown" can not only be common in firewalls, but also manifests itself in all sorts of middleboxes that rely on protocols and/or options being present or absent. This results in forbidding all protocols and packet types that are not commonly used by end systems behind the middlebox. The more we accept this as the common behavior and the more end systems rely on only the minimum everywhere-deployable element, the larger the "unknown" becomes. At the root of this problem is a circular dependency: what is not used is also useless, and hence can be blocked. In other words, a good reason to not allow new protocol X is that nobody demands it and no end systems regularly emit or expect packets of type X, making a packet of type X a strange, possibly dangerous, occurrence.

These obstacles to deployment are resulting in what has become known as "ossification".

However, it is important to realise the Internet is heterogeneous. Even if a certain protocol or a certain protocol mechanism is not supported on many paths of the Internet, this does not preclude it from being applicable and useful on *some* paths. Not every path has middleboxes that interfere with the transport layer to a degree that impacts evolution. Not even every path has a NAPT.

We therefore argue that Ossification is partly a result of the historical development process that has led to a range of transport protocols, but little consideration of how these relate to the transport layer itself. This has resulted in several alternate transports but with little or no planned support for applications wishing to use these in networks that only partly support them. We argue that end systems should try to use the protocols they want, to be able to benefit from paths that do support them, and efficiently fall back to a compromise behavior only in case a path does not support the desired protocol(s). This recommendation leads to another one as a consequence: since such try-and-fall-back functionality is complex to implement and may often not justify the potential gains for an application developer, it is important to not put the onus of having to develop such transport layer selection functions on the application developer. Rather, Operating Systems or libraries (end system middleware) should provide the required functionality – a set of mechanisms of general utility that can help to avoid re-inventing the wheel and being able to gradually improve for the benefit of all applications that use them.

The newly founded TAPS WG envisions a transport layer that can discover the transport protocols that are available. This can enable the Internet to become more robust (adapting to what transport protocols are supported by a path), and offer better performance (choosing a suitable protocol/mechanism to meet application needs).

References

[RFC 3205] K. Moore, "On the use of HTTP as a Substrate", RFC 3205 (Also BCP0056), February 2002.

[RFC6824] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

[1] M. F. Nowlan, N. Tiwari, J. Iyengar, S. O. Aminy, B. Ford, "Fitting square pegs through round pipes: unordered delivery wire-compatible with TCP and TLS", In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12), USENIX Association, Berkeley, CA, USA, 28-28, 2012.

[2] B. Briscoe, "Inner Space for TCP Options", Internet-draft draft-briscoe-tcpm-inner-space-01 (work in progress), October 2014.

[RFC 1191] J.C. Mogul, S.E. Deering, " Path MTU discovery", RFC 1191, November 1990.

[RFC 4821] M. Mathis, J. Heffner, " Packetization Layer Path MTU Discovery", RFC 4821, March 2007.