The ISP Column
A monthly column on things Internet

May 2021
Geoff Huston

Transport vs Network

One of the basic tools in network design is the so-called "stacked" protocol model. This model was developed in the late 1970s as part of a broader effort to develop general standards and methods of networking. In 1983, the efforts of the CCITT and ISO were merged to form The Basic Reference Model for Open Systems Interconnection, usually referred to as the Open Systems Interconnection Reference Model, or the "OSI model". This model included a seven-layer abstract model of networking that defined standard behaviours both of the overall network functionality but of the various components of the network.

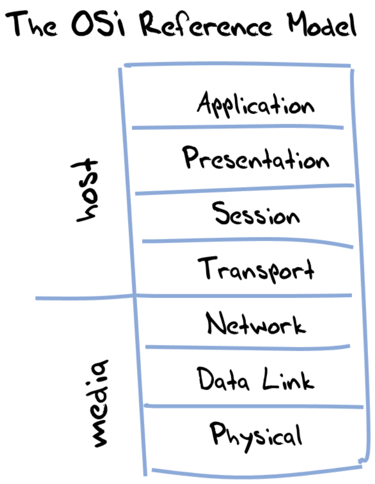

Figure 1 - OSI Reference Model

The model segmented functionality into two parts:

- the *media layers* that handle the encoding of binary data into the physical transmission media, the data link layer that describes the data frames used between two inter-connected "nodes" and the network layer that manages a multi-node network, including addressing and routing behaviours that manage the transmission of data between attached hosts.

- the *host layers* concern functions on end hosts. This encompasses the transport layer that performs data segmentation into packets, end-to-end flow control, packet loss recovery and multiplexing. The layers above the transport layer in the OSI model are the session layer, presentation layer and the application itself.

In a model of a network as a collection of interior nodes and a set of attached hosts, the interior nodes use only the network layer to make forwarding decisions for each packet that is handled, while the hosts use the transport layer to manage the data flow between communicating hosts.

The implication of this model is that there is a delineation between node and host functions and a clear delineation of the data that they need to perform their functions. Nodes do not need to have any knowledge of the settings used at the transport level, and, similarly, hosts have no need to access the network layer (Figure 2)

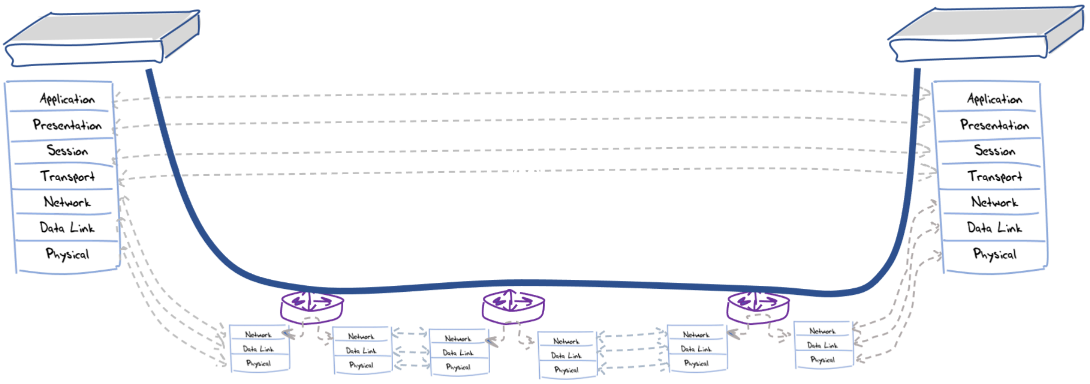

Figure 2 - Host and Node Functions

In the context of the Internet Protocol Suite, the network layer function is encoded as the IP header of a data packet, and the transport layer function is encoded as the Transport header, conventionally as either a TCP or a UDP header (although other headers are also defined by IP). In terms of the Internet architectural model, a packet should be deliverable through the Internet no matter what transport header it may have attached to it.

Therefore, it should not matter in the slightest what value you put in the IP protocol field in IP packet headers. It's really none of the network's business!

The same almost applies to the extension header fields in IPv6, although this is one area where, inexplicably, IPv6 scrambled the egg and some extension headers are addressed to network elements, namely the Hop-by-Hop and Routing extension headers, while the remainder are ostensibly addressed to the destination host. If Extension Headers were defined exclusively as host (or destination) extensions then the IPv6 networks should ignore them, while if they were intended to be network options then hosts should ignore them. Perhaps it's another of those areas where theory and practice just don't align well.

In a strict sense the Protocol field in the IPv4 packet header need never have been placed in the IPv4 header in the first place. The particular transport protocol being used by the communicating hosts is none of the network's business, in theory. This also means that if the two communicating hosts decide to deliberately obscure the transport protocol control settings from the network, then that should not matter in the slightest to the network.

In today's public Internet it appears to matter a lot that the transport protocol header is visible to the network. In fact, not only should the transport protocol be visible to the network, but the particular transport protocol selected by the hosts also matters to the network. The reason for this is that there are many elements of today's network that not only peek into the transport headers of the packets that they carry, but they also rely on the information in this transport header. Firewalls are a classic example of this reliance, but there are also Network Address Translators, Equal Cost Multi-Path load balancers, and Quality of Service policy engines, to name a few. These network functions make assumptions about the visibility of transport headers in the IP packet in order to make consistent decisions about packet handling for all packets within a single transport flow. Often these network functions take it one step further and they process packets with the well-known transport headers (typically restricted to just TCP and UDP) and discard all else. It's even gone futher than that, and we have reached the point that today's rule of thumb is that unfragmented IP packets that contain a TCP transport header that includes one end using port 443, and unfragmented IP packets that contain a UDP transport header where one end uses port 53 stand the best chance of getting their data payload through to the intended destination. Every effort to augment this remarkably constrained set of packet profiles increases the probability of network-based disruption of communication.

Encrypted Transport Headers

If it's a self-limiting action to use a novel transport protocol in the public Internet, then why are we even considering the option of encrypting transport protocols to make all transport headers opaque to the network?

One answer is "Edward Snowden." In response to these pervasive monitoring revelations [RFC 7624] the Internet Engineering Task Force (IETF) responded in what could be called a "like for like" reaction and came to a consensus position that "Pervasive Monitoring is an Attack" [RFC 7258]. The general response to this form of insidious attack was to increase the level of encryption of Internet traffic to lift the degree of difficulty in carrying out network-based surveillance. Not only does this IETF response encompass the use of TLS to encrypt session payloads on the Internet wherever possible and shift the application behaviour profiles to make this the default action, but also our attention has shifted to other areas of Internet communication where compromise of the trust model was thought to be an issue.

The actions of the DNS protocol have been drawn into this IETF universal obfuscation effort, as has the transmission of transport protocol headers. We are long past the time when hosts were ill-equipped to perform encryption functions and these days the use of robust encryption is not a luxury option with limited use, but something every user should reasonably expect to use as a minimum requirement. If the objective is to limit the information leakage in all aspects of the Internet's communications environment, then the control meta-data is as important as the data itself. Applying confidentiality to transport header fields can certainly improve users' privacy and can help to mitigate certain attacks or manipulation of packets by devices on the network path [RFC 8404].

However, I suspect that this privacy argument is only one part of the story, and while these measures to encrypt Internet traffic plays to a popular concern of the surveillance state operating in a largely unchecked manner, it may not lie at the heart of why obscuring host functions from the network is a path that some parts of the Internet ecosystem are vigorously pursuing today with transport header encryption.

It's not clear that the objective is here and as with all interdependent complex systems deliberately obscuring one aspect of the system from another is one that typically has both benefits and downsides. The IAB recently (April 2019) published RFC 8546, titled "The Wire Image of a Network Protocol". It's a short document (9 pages) by today's RFC standards, but brevity does not necessarily imply clarity. This document appears to have cloaked whatever message it was attempting to convey in such a dense level of abstract terminology that it appears to have managed to say absolutely nothing useful! The IAB document appears to have been prompted by the protracted debate in the QUIC Working Group over the use of the visible spin bit in the QUIC transport protocol (https://www.potaroo.net/ispcol/2018-03/onebit.html), and I suspect that it started as an effort to argue for some levels of transport behaviour visibility to the network, but the IAB's prognostications on the topic end up offering nothing in the way of useful or informative comment apart from illustrating the level of angst that this issue has generated! The IAB is not the most prolific of commentators, and any matter that provokes an IAB response, no matter how cryptic that response may be, does illustrate that the topic is one of general concern rather than a just being a rather esoteric tussle buried deep down in the design of a particular protocol.

This topic of encrypted transport headers is a transport topic, so it is natural to ask whether the IETF's Transport Area can do any better than the IAB in providing a clear and informed exposition of the issues here. The Transport Area Working Group of the IETF has completed a review of a draft on this topic, and the internet draft "Considerations around Transport Header Confidentiality, Network Operations, and the Evolution of Internet Transport Protocols" is now in the RFC Editor Queue (https://datatracker.ietf.org/doc/draft-ietf-tsvwg-transport-encrypt/).

To quote from document's abstract: "This document discusses the possible impact when network traffic uses a protocol with an encrypted transport header. It suggests issues to consider when designing new transport protocols or features."

This document strokes me as an effort to produce a slightly more practically focused commentary on header encryption than the earlier IAB effort. At 49 pages it certainly can't be considered a brief document, but does this extended commentary do any better in terms of clarity of the arguments being considered?

The document firstly looks are some rationales for the network's use of information contained in headers. They cite the situation of link aggregation and the problem of packet re-ordering is such scenarios. The common response to this is for the network to peer down into the transport header to gain a more granular view of a traffic flow than that which can be derived from source and destination IP address pairs. It's the IPv4 proxy for the IPv6 Flow label. (Although the IPv6 Flow label is so confused as to its intended role it's hard to understand how the IPv6 Flow Label field is useful in any case whatsoever!) The document references differential service efforts that attempt to perform selective damage on traffic flows under the guise of "Quality of Service" (That "Quality" label always seems to me to have an Orwellian connotation, and a more honest label would be "Selective Service Degradation", or even just "Carriage Standover Services"). The document also enumerates the ways network operators can perform network analysis of using transport level information, including traffic profile analysis, latency and jitter and packet loss. However, the document strikes me as presenting a somewhat disingenuous set of rationales. For me, it's akin to a voice telephony operator justifying its eavesdropping on phone conversations on the basis of a baseless assertion that that the information gathered by such wiretapping, or in other words knowledge of what people are saying to each over a telephone connection, can be used to make the telephone network better! The document also uses the last recourse of the desperate, by invoking a nebulous concept of "security", claiming that if network operators were no longer able to eavesdrop on the transport parameters of active sessions, then somehow the operator's ability to run a secure network would be compromised in some unspecified way.

Obviously, none of the rationales presented in this document can withstand much in the way of close scrutiny.

It also appears to take a privacy-oriented stance in its analysis, whereas it appears to me that the privbacy argument is largely an overt excuse for a more substantial difference of opinion between content and carriage. To a large extent the issue from the application's perspective is that the efforts of network operators to perform "traffic grooming" though transport header manipulation amounts to little more than inflicting damage on application data flows and thereby pushes the network to lower level of carriage efficiency. And this issue of the use of networks to selectively degrade transport performance in the name of network service quality is perhaps where we should look for the real tensions between networks and hosts in today's internet.

Transport Protocol Meddling

To look down this path we might want to start with the tensions between hosts and networks on the Internet.

In the telephone world the network operator controlled all traffic. What you leased from the network was either a virtual circuit capable to passing a real time voice conversation, or you could lease a fixed capacity channel between two end points. If you used one of these channels you couldn't go any faster than the contracted speed, and if you went slower then you did not release common capacity for anyone else to use. Obviously, the network charged more for leases of higher capacity. Packet networks changed all that. The network had no enforcement and various applications (or traffic flows) competed with each other for the common transmission resource. Networks that wanted to control the allocation of shared common communications resources to clients had a problem.

This was the motive for a large body of work on the Internet the 1990's and 2000 over what was called "Quality of Service" (or QoS). The network operator wanted to offer (no doubt for some premium) a “higher quality” service to some clients and some traffic profiles. But if a network has a fixed capacity offering a larger slice of the network’s resources to some clients inevitably means offering less to the others. One common theme of much of this work was that while it was possible for the network to disrupt a communication session in various ways to make it go slower, it was a lot more challenging (or even impossible in many ways) to make a session go faster. This meant that in order to offer preferential treatment to a class of traffic flows a good answer was to make all the other flows go slower! The intended effect was to clear some space for sessions that were intended to be favoured to expand their sending windows and occupy this cleared network space. So-called "Performance Enhancing Proxies" were not really able to make the selected TCP sessions go faster per se, but there were able to make other concurrent TCP sessions go slower, and thereby make some space for the selected sessions to have a lower packet loss probability and hence achieve a higher data throughput rate. One way of this form is session throttling is to drop packets. A more subtle way but also very effective is to alter the TCP control parameters. If the offered TCP window size parameter is reduced, then the sender will conveniently throttle their sending rate accordingly.

Pretty obviously, this selective behaviour of throttling active TCP sessions by networks was not something that applications viewed as a sympathetic act, and there have been two major responses from the application size. Firstly, there is the use of a different congestion control algorithm that is a lot less sensitive to packet loss and more sensitive to changes in the end-to-end bandwidth delay product across network paths. This is the BBR TCP control protocol, which is a relatively new TCP sender-side control algorithm. But BBR is still susceptible to on-path manipulation of the TCP window size and protecting the session from this form of network interference is where encrypted transport headers emerged became an important objective. This is the second response, namely by obscuring where the TCP control information is actually carried in the packet.

As we’ve already noted, you just can't remove a visible transport header from IP packets in the Public Internet, and even encrypting the TCP header would probably incur the same drop response from the network. But hosts have the option to ignore these transport header settings. So, while the host can’t remove a visible transport header, hosts can make them meaningless.

One option is to use a "dummy" outer TCP wrapper as fodder for networks who want to peek at transport and manipulate the session settings while hiding the real TCP control header inside an encrypted payload. There would be little in the way of a visible network signature that this is happening, apart from the observation that the TCP end hosts would appear to be unresponsive to manipulation of their window parameters.

However, the problem with this approach is that these days the application is actually trying not only to take control over its transport session parameters from a meddling network, but it’s also trying to assert the same control over the platform in which the application is hosted. In theory, the application could use "raw IP" interfaces into the platform's I/O routines, but in practice in deployed systems this is close to impossible. Platforms used in production systems tend to treat applications with suspicion. (Given the proliferation of malware this level of paranoia on the part of the platform is probably warranted.) It is quite a challenge to disable all forms of the platform's handling of the transport protocols and pass control of the transport protocol from the kernel into the applications space.

For this reason, it is logical to take the approach used by QUIC, where the shim wrapper of QUIC uses UDP as a visible transport header and pushes the TCP header into the encrypted payload part of the IP packet. UDP is close to ideal in this case as there are no transport controls in the protocol, just the local port numbers. QUIC looks to the network a lot like a UDP session that uses a TLS-like session encryption because in so many ways it is a UDP session that uses TLS. The change is that the end-to-end TCP flow control is now truly an end-to-end flow because only the two applications at the "ends" of the QUIC transport can see end-to-end transport control parameters that are embedded in the end-to-end encrypted UDP payload. The host platform control over UDP packets is perfunctory, and the application is then allowed to assume complete control over the session's transport behaviour.

Content vs Carriage

Perhaps this shift to opaque transport headers goes a little further than just a desire for greater levels of protected autonomous control by applications. The shift that QUIC represents could be seen as the counter move by content providers to another round of a somewhat tired old game play by networks operators to extract a tax from content providers by holding their content traffic to ransom, or, as it came to be known, a tussle over "network neutrality".

There have been times when network operators have implemented measures to throttle certain forms of traffic that they asserted was using their network in some vaguely unspecified manner that was "unfair" in some way. The vagueness of all this is probably attributable to a baser desire on the part of the carriage operator, which was to extort a carriage toll from content providers in a crude form of basic blackmail: "My network, my rules. You customer, you pay!"

I suspect that many carriage providers in this industry, who are witnessing the content providers take all the money off the table, believe that they are the victims here. Their efforts to restore some of their lost revenue base has meant that they are looking to restore a "fair share" of revenue in forcing the giants of the content space to pay for their share of carriage costs. However, if the enforcement mechanism of this extortion pressure is through playing with the transport control parameters of the traffic that transits the carriage network (or, in other words, holding the traffic to ransom), then the obvious response is to push the transport controls under the same encryption veil as the content itself to prevent such on-the-fly manipulation of the traffic profile. And this is perhaps a more compelling explanation of why QUIC is so important.

If this is a tussle for primacy in the tensions between carriage and content, then it sure looks like the content folk are gaining the upper hand. Through encryption at every level in the host part of the protocol stack, including at the transport layer, the content folk are withholding information from the carriage providers that would allow the carriage provider to selectively discriminate and play content providers off against each other. If all that the network can is limited to fully encrypted UDP packet streams, then one stream looks much like another and selective discrimination is just not feasible. And if that’s not enough then padding and deliberate packet variation can blur most efforts at traffic profiling.

But when I say “content” I really mean “apps” and when I say “apps” I actually mean “browsers”, so in reality I am really talking about Chrome, and when I say Chrome, I mean Google.

The massive dominance of mobile traffic in the industry and the massive dominance of Android in the mobile device environment tilts this space to an extraordinary degree. Given this inherent level of control of all mobile devices, coupled with control of the majority browser platform in this space it is hard to conceive how Google could possibly lose in this tussle. However, it is unlikely that once Google win this particular battle with the carriage providers, there won’t be further battles to come. It is highly likely that the carriage industry will follow the lead from traditional print media and head to politicians with the case that Google’s destruction of the business model for the provision of national communications infrastructure is counter to national interests, and political intervention is necessary to restore some balance into the market and allow the market for carriage to be a viable investment vehicle. Or, to put in more crudely, if Google has destroyed the residual value contained carriage market, then Google should now pay carriage operators to restore its viability.

At this point all technical considerations of encryption and information leakage, and even all market considerations of the viability of various business models just walk out the door, and in their place comes a bevy of lawyers and politicians. Strategic national interest is always a strong card to play, and once we get over the various nebulous threats by actors to quit national markets, we then get down to the real question of: “What is a tenable business relationship between carriage and content?”

In such a politically charged space the choices at that point are either that the various market players will compromise and reach some outcome that they can all live with, or the politicians will attempt to impose an outcome that will in all likelihood be far more disagreeable for all!

Whatever the outcome in the next few years it should be fun to watch this play out. Don't forget to bring popcorn!

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net