The ISP Column
A monthly column on things Internet

Geoff Huston
July 2016

Hosts vs Networks

There are a number of ways to view the relationship between hosts and the network in the Internet.

One view is that this is an example of two sets of cooperating entities that share a common goal: hosts and the network both want content to be delivered. Both have an interest in seeing this delivery happen quickly, efficiently and without fuss. This leads to one view of the relationship between hosts and the network: the more the elements of the network provide the host with a description of their capabilities and connectivity, and the more the host provides the network with a description of the desired treatment of their traffic flow, the more harmonious the relationship, and the better the carriage service can attune its service response to match the host's requirements, and vice versa.

However, there is a second view of this relationship. One which sees this engagement defined by conflicting objectives. For example, the network is attempting to maximize its ability to extract money from the provision of carriage services, while the host application is attempting to maximize its use of the carriage resource while minimizing its monetary expenditure. Or a Carrier Grade NAT (CGN) in the network may be attempting to share a limited pool of addresses and ports over as large a set of sessions as possible, while each session may want a stable NAT state to last as long as the session

At one point in time it was believed that as cooperative stance between networks and hosts could lead to a better service outcome. Hosts wishing some superior level of service could signal this to the network and have their traffic treated in some preferential manner, and out of this came the concept of Quality of Service (QoS). Despite some iterations over the exact nature of the QoS architecture, none of the QoS approaches enjoyed widespread deployment, and their use today remains largely as a niche service with some possible application in an enterprise environment, but not elsewhere. So, in some sense the model of a mutually cooperating set of entities working to a common purpose appears to be something seen more commonly as possibilities written up in specifications than in public utility networks.

But that observation shouldn't lead you to believe that networks don't actively manipulate the profile of the traffic that passes across it. Traffic classifiers, shapers, interceptors, proxies, NATs and other forms of active middleware abound, and if anything their use is even more widespread today then earlier. The critical difference here is that such network responses are not carried out at the behest of the host and signalled via explicit settings in its traffic flow, but instead the network imposes these responses on the traffic it carries. This is more like the second view outlined above, where the network's objectives and those of the host appear to be in conflict rather than in alignment.

In this world it is not necessarily in the interests of the host to present traffic to the network in a manner that exposes its flow parameters. All that such exposure would achieve would to broaden the set of opportunities for the network to perform manipulation of traffic. So rather than expose the path and flow characteristics to the network, a host may take the stance that its best interests, and those of the user behind the host, are served by cloaking its traffic flow parameters from the network.

These two perspectives were clearly evident for me at the recent IETF 96 meeting, where there were two Birds of a Feather (BOF) sessions about using UDP as an end-to-end transport service substrate. One approach is typified by Google's QUIC, which hides its control parameters from the network and leaves only a minimal information set exposed in the unencrypted header part of the UDP packet. The other was the BOF session on PLUS, which, to quote from a description provided in the BOF, is "A mechanism for making widespread cooperation between endpoints and middleboxes explicit."

QUIC

The QUIC BOF described the development of QUIC in Google. The motivation here is that the proliferation of active TCP-ensnaring middleware in the network has reached the point where not only is further innovation in the flow control signalling in TCP generating no further corresponding increases in carriage efficiency, but that today's middleware is actively working against the host, and altering the TCP flow management mechanisms to suit their own objectives at the cost of the end-to-end application.

In some ways QUIC is a very simple adaptation. QUIC uses the platform's UDP substrate rather than the provided TCP service. IP Packets are loaded with protocol value 17 (UDP) rather than 6 (TCP) and the following 8 octets contain UDP port numbers, the UDP packet length and the UDP packet checksum. But at this point there is a change to a conventional use of UDP. There a TCP-like implementation sitting within the application, and it uses the UDP datagram transport in the same way that conventional TCP uses the IP datagram transport.

The downside of this TCP-in-UDP approach is that packets are 8 octets longer, and of course QUIC hosts can only communicate with other QUIC hosts, so unlike TCP this is not an instance of a new end-to-end transport lingua franca for the Internet. But this is a case of what you may lose at the roundabouts you can gain on the swings, and in this case the gain is all about the ability to run an end-to-end protocol that is not only free from interference from network middleware, but also this approach restores clear and coherent end-to-end signalling, allowing the user-level implementation of the end-to-end transport some considerable room to innovate with novel flow management approaches.

The BOF was focused on the potential for innovation in end-to-end flow management protocol, and the way in which a number of recently considered innovations could be integrated into the flow management system.

It should also be noted that this cloaking of the inner TCP flow state from the network is not just by virtue of using the inserted UDP header. QUIC also encrypts its payload, so that the signaling within QUIC is not exposed to middleware, even if they were to be aware of QUIC signaling within the packet.

The BOF was one of the more positive BOFs in recent times: there was running code plus documentation and reports of implementations from Microsoft and Akamai as well as the original work by Google. There was also a report on performance returns, with an average of 5% faster page load times when using a QUIC substrate. So we have a protocol that has running code, performance returns, NAT agility, security and flexibity. No wonder that the BOF generated a strong positive signal from the attendees to press on with this work.

PLUS

The PLUS BOF certainly struck a different tone from this IETF meeting than one would've encountered a decade or so ago. I suppose that this sensitivity has occurred after being made painfully aware that any form of openness of Internet traffic in the network is being exploited by third parties. RFC7258 advanced the proposition that "Pervasive monitoring is a technical attack that should be mitigated in the design of IETF protocols, where possible", and it's challenging to make the case that PLUS would not assist monitoring of UDP flows. One of the consequences of this exhortation to mitigate the opportunities of monitoring is to view any form of information leakage of user traffic, or even the control profile of user traffic, as a potential opportunity for potentially hostile monitoring. So rather than looking at PLUS as an opportunity to allow the network to gain control profile information from the application traffic flow so that it can provide a superior overall service response to all flows and all users, the common view of PLUS appears to have strongly negative aspects. These days information exposure is seen as unnecessary, and any attempts by the network to alter the user traffic flow in any way is also regarded in a negative light.

PLUS and QUIC are similar in one aspect: both are end-to-end transport protocols that are intended to operate as overlays across a UDP transport. But at this point the two approaches rapidly diverge.

What does PLUS propose to make explicit that a directly comparable transport protocol such as QUIC does not? Perhaps one area is the potential for explicit session signaling. NATs use the opening tCP SYN handshake to create a NAT binding, and, within reasonable bounds, they keep the binding open until a TCP FIN or a TCP RST occurs. UDP generically provides no such explicit session signalling. The initial outgoing packet creates a NAT binding, and the binding is maintained with an idle timer, often cited to be 30 seconds (although measurement studies are hard to come by). PLUS-aware NATs would bring explicit flow start and stop signalling back into this picture, allowing the NAT state to be aligned to the underlying application behaviour through the use of network-visible PLUS flow start and flow stop signals. It also has areas of potential use in the area of subflow control over multiple network paths. Explicit signaling to the network about which parts of a flow form part of a subflow and which elements have a strong preference to preserve relative packet order could assist the efficiency of the application treatment of the subflow within the network.

In looking at how to implement this desired functionality, one approach is to combine PLUS exclusively with IPv6 and use the IPv6 Flow Identifier field and the IPv6 Extension Headers as the place for control signaling for flows and subflows. As interesting as this sounds in theory, a sobering observation was recently published as RFC7872, which pointed out that today's network exhibited unacceptably high packet drop rates when Extension Headers are added to an otherwise viable IPv6 end-to-end connection. It seems that many deployed IPv6 routers take exception to the presence of any form of Extension Header in IPv6 packets. So either we need to be very patient with this type of approach and wait for the Extension Header drop problem to be comprehensively addressed by equipment vendors and network operators, and have a fully functional set of hop-by-hop Extension Headers that signal flow and subflow behaviours implemented at wire speed, or we need to look elsewhere. Another approach is to devise a new field in PLUS, that acts as a UDP session/flow shim layer. This also poses some issues, including the consequence that active middleware needs to be altered to look for this new control parameter block in every UDP packet, which presents its own deployment challenges.

PLUS is at a far earlier state than QUIC in terms of specification and implementations, and even in terms of proponents. Aside from a coterie of what appears to be rusted-on IETF attendees, it seems that the major backers of this proposal are the mobile data carriers. They see the evolving moves involving cloaking of the upper layer protocols from the carriage system as an anathema, as they have evolved their services with a business model that relies on tight control over content passed across their network.

The reaction of the IETF at the BOF was not exactly hostile, but it was clear that there was much resistance in the room to standardizing this approach to a transport protocol. It appears that our patience is running thin with protocols that emit readable metadata. It's not even clear that at this point in time there is any major appetite in the IETF for working on transport protocols that are intended to leak information to facilitate the operation of network middleware.

Time for the Paranoid Application?

It seems that the Internet has turned another corner in the past few years. It now appears that there no further impetus for designing systems where there is open cooperation between networks and the applications that run on hosts. The implicit mutual trust of the early Internet period has all but disappeared, and applications increasingly regard the network as a hostile environment that should be negotiated with care and discretion. This paranoia on the part of some application designers may go further. It's not unusual to hear of an application that eschews the traditional common platform services of DNS name resolution and the TCP transport protocol. Not only does this give the application far greater control over the application's service, but it can be used to ensure that the application runs with integrity. With appropriate measures the application can ensure that neither the network nor the host on which the application is running can readily snoop on the application. If this is coupled with TLS at the transport level, then the application can gain some assurance that it is not the subject of data tampering and attempts to subvert its operation.

We've come a long way from a world of open protocols and an environment built upon mutual trust. We've becomes jaundiced by repeated forms of abuse of this openness and trust, and increasingly we are seeing a more defensive posture taking its place. Applications are now wary of the environment in which they are operating, and all the evidence appears to suggest that there is good cause to exercise such caution.

So when it comes to UDP session control design it's not surprising to see PLUS attract a cautious, or even somewhat hostile, reception at the IETF. QUIC is a more eloquent summation of our current world. Pull everything back behind the shell of the application, and encrypt the data flow as it leaves the application to ensure that no one else can clearly see, let alone interfere, with the data content.

This may seem like paranoia on the part of the application, but just because you're paranoid it doesn't mean that the bastards aren't out to get you!

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990's. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001 and chaired a number of IETF Working Groups. He has worked as an Internet researcher, as an ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer

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