Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2013

H. Tschofenig
Nokia Siemens Networks

July 9, 2012

The New Waist of the Hourglass
draft-tschofenig-hourglass-00.txt

Abstract

When developing a new application layer protocol the question about its foundation arises. The decision will be impacted by various factors but the ability for the solution to be deployable in today's Internet will certainly play a big role since various intermediaries may lead to a brittle communication architecture.

The waist of the hourglass has changed over time: new applications have to run on top of UDP or TCP. Protocol designs that use other transport protocols have turned out to be undeployable on the public Internet. Running protocols on bare IP is also doomed to fail. This change is not theoretic in nature but has real-world implications for protocol designers.

There is also a trend to run applications on top of HTTP/HTTPS. The author seeks more input from the wider IETF community about the deployment situation of protocol filtering by intermediaries and on the impact for protocol designers.

This document is being discussed at
https://www.ietf.org/mailman/listinfo/architecture-discuss

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2013.

Table of Contents

1. Introduction

The Internet protocol stack is organized in layers and text books typically talk about the data link layer, network layer, transport layer, and the application layer.

IP evolved in a core part of protocol stack that represents a common intermediate protocol layer. This intermediate layer is designed to support a variety of link layers and new link layer technologies can be added in the future, without affecting IP itself.

In addition, IP supports a broad set of applications on top of it, ranging from email to instant messaging to voice over IP. All of those applications do not require changes to IP itself. As with link layers, new applications can be added at any time.

The "Everything over IP, and IP over everything" theme, as it is often described, is shown graphically in Figure 1.

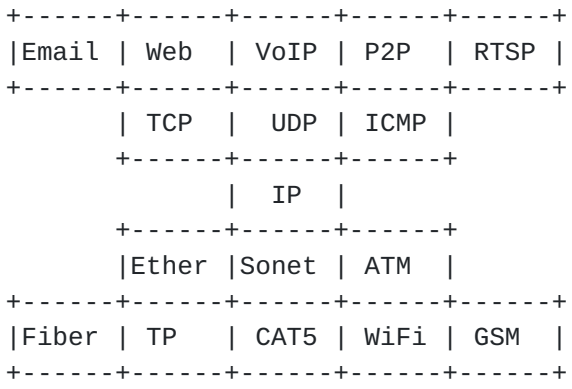

Figure 1: The Original IP Hourglass

This diagram is, of course, simplified but illustrates the level of interoperability needed to communication between different IP hosts. Combined with the end-to-end principle the hour glass indicates the level of protocol understanding intermediaries need to have in order to exchange forward IP packets between a sender and a receiver (absent any specific application layer entities, like proxies or caches). Having IP as the waist meant that anyone could extend the layers above the network layer in the way they wanted to communicate end-to-end, including defining new transport layer protocols (as it was done with SCTP, and DCCP).

Unfortunately, the various developments over the last years have changed this model. We will describe the developments below.

2. Network Address Translators

Originally, NAT was designed to operate strictly at the IP layer - translating internal to external addresses 1-for-1. This was done primarily to avoid network renumbering when there was a change in provider. However, NAT-P - NAT with port translation - quickly emerged. In NAT-P, a large number of hosts can utilize a single external IP address by using the UDP and TCP port numbers as a demultiplexing point.

In essence, the UDP or TCP port number became an extended version of the IP address - adding another 16 bits to the total amount of space, providing for 48 overall.

NAT-P in particular, often just called NAT, has seem extremely widespread deployment. Needless to say that this is a common deployment in residence with broadband Internet but with the work on Carrier Grade NAT it has also become a solution for dealing with IPv4 shortage in their effort to migrate to IPv6.

When put together, the implication is that basic packet transport between point A and point B is, these days, frequently possible only if the transport is UDP or TCP. This is because NAT-P requires one of these two in order to utilize the 16 bit ports as as a demultiplexing point. NATs, which are part of the network, have become not just TCP and UDP-aware, they have become TCP and UDP *dependent*. Almost every NAT in deployment today will simply discard a packet above IP that is not TCP or UDP. The primary exception is ICMP. Some NATs will pass ICMP packets, but it is filtered by many. Consequently, it only sometimes works on the Internet. This is also making it difficult to rely on. Indeed, it's success rate is sufficiently poor that new mechanisms for path MTU discovery have been designed which work without ICMP [RFC4821].

This means that the operation of the public Internet is dependent on the existence of UDP and TCP traffic.

Consequently, if new protocols have to ever actually work on the public Internet, they need to run on top of the only available packet transport on the Internet - UDP/IP. UDP has replaced IP as the 'raw' point to point packet transport on the Internet. If congestion control or signaling or other features are desired, they must be layered on top of UDP, rather than in a new protocol beside it.

The IAB had published a number of publications to make the community aware of the implications. In RFC 3724 the IAB examines the development of the end-to-end principle as it has been applied to the Internet architecture. In RFC 3424 [RFC3424] the IAB specifically

addressed the problems created with NATs.

3. New Transport Protocols

Does this mean that it is impossible to define new transport protocols? Fortunately, the answer is no. New transport protocols can be defined but the have to build on top of UDP or TCP. The considerations for running them on top of UDP/TCP from Section 2 apply.

It is tempting to design a protocol on top of IP in such a way that it is "NAT friendly". In this context, "NAT friendly" means that the protocol has been designed to make ALGs for that protocol easy to implement. However, there is a vicious cycle that prevents such ALG functionality from ever being built. New features get added to products, such as NATs, when the market demands them. The market demands them when there is enough usage to create such a demand. However, if the protocol will fail utterly to work in the face of existing NAT, there will never be enough usage to create such demand. Even if there was, the amount of time it would take to upgrade all of the NATs on the Internet to support this ALG functionality is astoundingly large. Thus, the protocol will continue to be unreliable on the Internet, since there is always a chance that it hits a NAT which does not support the necessary ALG functionality.

Consequently, designing protocols to be "NAT friendly" in this way does not work in practice. New applications and protocols must be designed to run on top of either UDP or TCP.

Examples of where this has been done after the fact is [I-D.ietf-tsvwg-sctp-udp-encaps] for SCTP and [RFC3948] for IPSec. However, it is far better to define this at the beginning, and furthermore, have it as the one and only mode of operation. This avoid protocol choices, and therefore simplifies design and improves interoperability.

4. Firewalls

Unfortunately, it is not just NATs that cause problems for end-to-end communication. Firewalls are also TCP and UDP aware, and are often configured to discard non-UDP or non-TCP traffic. The degree of filtering depends on the type of network; enterprise networks have traditionally been more aggressive in filtering even outgoing traffic.

Many firewalls are stateful, i.e., they create state information when they see an packet from the internal network towards the Internet. An unpleasant property of stateful packet filtering firewalls is that they may lead to dropped packets when an arriving packet does not match the previously allocated state information or when they drop packets as part of their policy (as it is often the case when non-UDP/non-TCP packets are seen). Furthermore, the firewall (as well as NATs) decide locally when to time-out state information without informing any other network entities leaving end hosts guessing about the rate at which they need to send keep-alive messages. Time-out intervalls typically varry between TCP and UDP but vary considerable between different deployments and between products from different vendors.

IETF protocol designers have already revised some protocols to handle stateful packet filtering firewalls. Examples can be found with RFC 4487 [RFC4487] and RFC 5207 [RFC5207].

Clever protocol designers started to invent algorithms to test paths between address pairs for reachability. These patch tests are time consuming and need to be re-run regularly when the network conditions change. The more address pairs are available (IPv4, IPv6, tunnels, etc.) the longer the tests need.

One such path test mechanism is the Interactive Connectivity Establishment (ICE) protocol, described in RFC 5245 [RFC5245], and it has been successfully used with the Extensible Messaging and Presence Protocol (XMPP) and the Session Initiation Protocol (SIP). Similar mechanisms may also be re-used with the efforts around the Real-Time Communication in WEB-browsers, which is currently work in progress in the IETF and the W3C.

For those cases where a protocol aims to exchange traffic between hosts where both sides are behind NATs or firewalls then a rendezvous mechanism and a mechanism for patch tests is needed. Such a mechanism, however, also allows to determine when fewer protocol encapsulations can be used.

To combat the ever increasing number of path candidates that have to

be tested for real-time applications a new protocol design practice had emerged: Instead of using separate port pairs for different media streams (e.g., streams for voice and video RTP traffic, feedback for RTP via RTCP, separate channel for signaling communication and inband communication, e.g., instant messaging, key exchange protocols) protocol architects design their protocols in such a way that they only use a small set of UDP/TCP port pairs and multiplex/demultiplex the application traffic at a higher layer. This reduces the number of path candidates that have to be tested for e2e connectivity.

This design trend can already be observed since a few years with examples being found with DTLS / SRTP. RFC 5764 [RFC5764] and RFC 5763 [RFC5763] describe the design and how to demultiplex arriving packets. The level of multiplexing is astonishing:

- Symmetric RTP [RFC4961] is the case in which there are two RTP sessions that have their source and destination ports and addresses reversed.

- RTP and RTCP traffic is usually sent on two separate UDP ports. Two bidirectional DTLS-SRTP sessions are needed when symmetric RTP [RFC4961] is used: one for the RTP port, one for the RTCP port.

- RTP and RTCP traffic may also be multiplexed on a single UDP port [RFC5761].

- Finally, the DTLS, and the connectivity check packets using STUN also need to be multiplexed over the same port pair to accomplish best possible optimization.

This improves the cryptographic performance of DTLS, but may cause problems when RTCP and RTP are subject to different network treatment (e.g., for bandwidth reservation). Some organizations using SIP as part of their communication architecture have made assumptions about what network intermediaries can do in order to perform firewall pinholing and QoS treatment that may consequently not be compatible with this type of design anymore. See [I-D.ietf-mmusic-media-path-middleboxes] for a discussion about middlebox behavior with SIP in context of media path security.

5. Will IPv6 reverse the shift?

Will the IPv6 Internet share the same fate as the IPv4 Internet, and work only with UDP and TCP? It seems likely that the answer will be yes.

The primary reason is that the IPv6 Internet is not something that will appear overnight to replace the IPv4 Internet, as everyone had noticed by now already. It will run alongside it for a very long time, if not forever. Hosts that have connection to the IPv6 Internet will find themselves frequently using IPv4 (in a dual- stack deployment), because the target host is available only on IPv4, or will find themselves communicating with via IPv6 to a v4-only host through NAT. In both cases, any protocols except for UDP and TCP based protocols will not work. And thus, the v6 host will need to utilize protocols that do work in all cases - ones based on UDP and TCP - rather than ones that work only in a few cases. And so, when we eventually do cutover to IPv6 only, it will be with hosts which have, all along, only been running protocols that run on top of UDP or TCP.

Another reason is that the IPv6 Internet will certainly be filled with firewalls, and if history is any guide for the future, only TCP and UDP are likely to work through such firewalls.

Examples of such firewall guidance for IPv6 to mimic the security functionality of a NAT in IPv6 is given in RFC 4864 [RFC4864].

Furthermore, (too) many efforts are currently ongoing to deal with the long transition period from IPv4 to IPv6 and these migration techniques introduce additional layers of NATs and tunneling that will lead to problems with bi-directional reachability.

Finally, the IPv6 Internet is filled with NATs already anyway, despite attempts to provide ample address space.

Consequently, for an application designers perspective, why build an application on top of a protocol which doesn't work on the IPv4 Internet, and won't work on anything but a pure IPv6 network, when an application running on top of UDP or TCP will work everywhere?

6. The Rise of the Web

The Web had enjoyed incredible success over the last 20 years and so it is not suprising that many application developers have used HTTP and HTTPS as the foundation for their application protocol design. The properties offered by HTTP, HTML, and JavaScript are often a good fit for many designs, and the familiarity of HTTP appeals to many developers. With the intensive standardization efforts that happened over the last few years and the improvement in browser peformance Web applications today offer the same capabilities as native applications in almost all areas. More guidance when HTTP usage is appropriate can be found in [I-D.tschofenig-http-design-guidance].

Sometimes, however, the choice of HTTP is not entirely voluntarily. Certain networks, specifically enterprise networks, deploy aggressive packet filtering technology such that barely any traffic other than HTTP/HTTPS is allowed to bypass. Whether this policy decision is based on a careful security evaluation, lack of awareness of other protocols, or too complex company-internal processes for interacting with the network administrator for updating packet filtering rules to allow new applications to traverse is difficult to say.

In any case, for applications that are intented to be used in such environments any protocol other than HTTP (or even HTTPS) is less likely to function. It is therefore not suprising that protocol designers are not willing to take the risk or put the additional complexity of supporting multiple transport mechanisms. Consequently, the perceived firewall traversal capability of HTTP/HTTPS has "convinced" many developers to use HTTP/HTTP as their protocol of choice. As proxies that perform their policy at the HTTP layer become more common application developers will have to react again: an classical arms race. Consequently, HTTPS becomes the only choice to prevent intermediaries from inspecting (and potentially modifying) application layer content.

While many protocol designs use HTTP/HTTPS the Web is, however, much more. In [I-D.tschofenig-post-standardization] we argue that the features provided by JavaScript, as used by many Web developments, have created another degree of sophistication that has helped to build a convenient platform for application development. Especially for end-user facing applications the JavaScript-based Web platform has changed the standardization landscape and the work of protocol architects.

7. Community Input Needed

The developments in the area of transport protocols have been recognized by the community for a couple of years now and NAT and firewall traversal techniques are taken into consideration in protocol designs..

The author is, however, struggling with the question whether there is enough evidence to conclude that every protocol design today shall build on HTTP/HTTPS (rather than voluntarily using to use HTTP/HTTPS because of its properties). This would lead to another shift in the IP hourglass model with HTTP/HTTPS being the new waist.

At this point in time it may be premature to conclude that a mandatory HTTP/HTTPS based protocol design is indeed appropriate.

Firewalling behavior certainly differs from environment to environment and not every Internet protocol is supposed to be deployed everywhere. Some protocols are specifically focused on server-to-server communication or have historically been using a non-HTTP/HTTPS based design (e.g., routing protocols).

The author is therefore seeking input and further research on this issue from the wider IETF and Internet community.

A few publications exist, such as [IPoptions], [TCPextensions], and [HomeGateway], that illustrate the degree of filtering in networks used by end customers (e.g., home networks, hotspots, and ISP networks). Note that enterprise networks are explicitly excluded since they are known to be challenging from filtering point of view and cannot serve as the basis for protocol design. There are also projects ongoing that may build the foundation for additional insight, such as the 'Open Observatory for Network Interference' [OONI] or the Neubot project [Neubot].

In any case, more data points are needed to offer a better snapshot of the current Internet deployment status. It will help protocol developers to make informed decisions for their designs.

8. Security Considerations

This document focuses on implications for protocol design and security protocols themselves are also impacted by the developments on the Internet caused by evolving behavior of network intermediaries.

9. IANA Considerations

This document does not require action by IANA.

10. Acknowledgements

The author would like to thank Jonathan Rosenberg for his work on [I-D.rosenberg-internet-waist-hourglass]. This document re-uses text from his earlier work.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

11.2. Informative References

[RFC4487] Le, F., Faccin, S., Patil, B., and H. Tschofenig, "Mobile IPv6 and Firewalls: Problem Statement", RFC 4487, May 2006.

[I-D.tschofenig-post-standardization] Tschofenig, H., Aboba, B., Peterson, J., and D. McPherson, "Trends in Web Applications and the Implications on Standardization", draft-tschofenig-post-standardization-02 (work in progress), May 2012.

[I-D.rosenberg-internet-waist-hourglass] Rosenberg, J., "UDP and TCP as the New Waist of the Internet Hourglass", draft-rosenberg-internet-waist-hourglass-00 (work in progress), February 2008.

[RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.

[I-D.ietf-mmusic-media-path-middleboxes] Stucker, B., Tschofenig, H., and G. Salgueiro, "Analysis of Middlebox Interactions for Signaling Protocol Communication along the Media Path", draft-ietf-mmusic-media-path-middleboxes-04 (work in progress), January 2012.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol

(SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, May 2010.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5207] Stiemerling, M., Quittek, J., and L. Eggert, "NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication", RFC 5207, April 2008.

[I-D.ietf-tsvwg-sctp-udp-encaps]
Tuexen, M. and R. Stewart, "UDP Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-udp-encaps-04 (work in progress), July 2012.

[RFC3724] Kempf, J., Austein, R., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, March 2004.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC3424] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[I-D.tschofenig-http-design-guidance]
Tschofenig, H., Dusseault, L., Reschke, J., and M. Nottingham, "HTTP Protocol Design Guidelines", status: not yet published, July 2012.

[Neubot] Nexa Center for Internet & Society, "The Neubot Project", available at: http://nexa.polito.it/neubot, 2012.

[OONI] The Tor Project, "OONI : Open Observatory of Network Interference", available at: http://ooni.nu/, 2012.

[HomeGateway]
Eggert, L., "An experimental study of home gateway characteristics, In Proceedings of the '10th annual conference on Internet measurement'", 2010.

[TCPextensions]
Honda, M., Nishida, Y., Greenhalgh, A., Handley, M., and H. Tokuda, "Is it Still Possible to Extend TCP? In Proc. ACM Internet Measurement Conference (IMC), Berlin, Germany", Nov 2011.

[IPoptions]
Fonseca, R., Porter, G., Katz, R., Shenker, S., and I. Stoica, "IP options are not an option, Technical Report UCB/EECS", 2005.

Author's Address

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at