Out With the Old and In With the New: Planning for Protocol Transitions
draft-iab-protocol-transitions-03.txt

Abstract

Over the many years since the introduction of the Internet Protocol, we have seen a number of transitions from one protocol or technology to another, throughout the protocol stack. Many protocols and technologies were not designed to enable smooth transition to alternatives or to easily deploy extensions, and thus some transitions, such as the introduction of IPv6, have been difficult. This document attempts to summarize some basic principles to enable future transitions, and also summarizes what makes for a good transition plan.

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 March 16, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

A "transition" is "the process or period of changing from one state or condition to another". There are several types of such transitions, including both technical transitions (e.g., changing protocols or deploying an extension) and organizational transitions (e.g., changing what organization manages the IETF web site, or the RFC production center). This document focuses solely on technical transitions, although some principles might apply to other types as well.

There have been many IETF and IAB RFCs and IAB statements discussing transitions of various sorts. Most are protocol-specific documents about specific transitions. For example, some relevant ones in which the IAB has been involved include:

- IAB RFC 3424 [RFC3424] recommended that any technology for so-called "unilateral self-address fixing (UNSAF)" across NATs include an exit strategy to transition away from such a mechanism. Since the IESG, not the IAB, approves IETF documents, the IESG thus became the body to enforce (or not) such a requirement.
IAB RFC 4690 [RFC4690] gave recommendations around internationalized domain names. It discussed issues around the process of transitioning to new versions of Unicode, and this resulted in the creation of the IETF Precis WG to address this problem.

The IAB statement on "Follow-up work on NAT-PT" [IabIpv6TransitionStatement] pointed out gaps at the time in transitioning to IPv6, and this resulted in the rechartering of the IETF Behave WG to solve this problem.

More recently, the IAB has done work on more generally applicable principles, including two RFCs.

IAB RFC 5218 [RFC5218] on "What Makes for a Successful Protocol?" studied specifically what factors contribute to, and detract from, the success of a protocol and it made a number of recommendations. It discussed two types of transitions: "initial success" (the transition to the technology) and extensibility (the transition to updated versions of it). The principles and recommendations in that document are generally applicable to all technical transitions. Some important principles included:

1. Incentive: Transition is easiest when the benefits come to those bearing the costs. That is, the benefits should outweigh the costs at "each" entity. Some successful cases did this by providing incentives (e.g., tax breaks), or by reducing costs (e.g., freely available source), or by imposing costs of not transitioning (e.g., regulation), or even by narrowing the scenarios of applicability to just the cases where benefits do outweigh costs at all relevant entities.

2. Incremental Deployability: Backwards compatibility makes transition easier. Furthermore, transition is easiest when changing only one entity still benefits that entity. In the easiest case, the benefit immediately outweighs the cost and so entities are naturally incented to transition. More commonly, the benefits only outweigh the costs once a significant number of other entities also transition. Unfortunately, in such cases, the natural incentive is often to delay transitioning.

3. Total Cost: Don’t underestimate the cost of things other than the hardware/software itself. For example, operational tools and processes, personnel training, business model (accounting/billing) dependencies, and legal (regulation, patents, etc.) costs all add up.
4. Extensibility: Design for extensibility so that things can be fixed up later.

IAB RFC 7305 [RFC7305] reported on a IAB workshop on Internet Technology Adoption and Transition (ITAT). Like RFC 5218, this workshop also discussed economic aspects of transition, not just technical aspects. Some important observations included:

1. Early-Adopter Incentives: Part of Bitcoin’s strategy was extra incentives for early adopters compared to late adopters. That is, providing a long-term advantage to early adopters can help stimulate transition even when the initial costs outweigh the initial benefit.

2. Policy Partners: Policy-making organizations of various sorts (RIRs, ICANN, etc.) can be important partners in enabling and facilitating transition.

The remainder of this document continues the discussion started in those two RFCs and provides some additional thoughts on the topic of transition strategies and plans.

2. Transition vs. Co-existence

There is an important distinction between a strict "flag-day" style transition where an old mechanism is immediately replaced with a new mechanism, vs. a looser co-existence based approach where transition proceeds in stages where a new mechanism is first added alongside an existing one for some overlap period, and then the old mechanism is removed at a later stage.

When a new mechanism is backwards compatible with an existing mechanism, transition is easiest, and the difference between the two types of transition is not particularly significant. However, when no backwards compatibility exists (such as in the IPv4 to IPv6 transition), a transition plan must choose either a "flag day" or a period of co-existence. When a large number of entities are involved, a flag day becomes impractical. Coexistence, on the other hand, involves additional costs of maintaining two separate mechanisms during the overlap period which could be quite long. Furthermore, the longer the overlap period, the more the old mechanism might get further deployment and thus increase the overall pain of transition.

Often the decision between a "flag day" and a sustained co-existence period may be difficult, such as in the case of IDNA2008 [RFC5891] [RFC5895] and Unicode TR46 [TR46].
3. Translation/Adaptation Location

A translation or adaptation mechanism is often required if the old and new mechanisms are not interoperable. Care must be taken when determining where such a translator is best placed.

Requiring a translator in the middle of the path can hamper end-to-end security and reliability. For example, see the discussion of network-based filtering in [RFC7754].

On the other hand, requiring a translation layer within an endpoint can be a resource issue in some cases, such as if the endpoint could be a constrained node [RFC7228].

Any transition strategy for a non-backward-compatible mechanism should include a discussion of where it is placed and a rationale.

4. Translation Plans

The case studies described in Appendix A suggest that a good transition plan includes at least the following components:

1. An explanation of incentives for each entity involved
2. A description of transition phases. For example, there might be pilot, co-existence, deprecation, and removal phases for a transition from one technology to another incompatible one.
3. A proposed timeline
4. A way to measure whether the transition is succeeding
5. A contingency plan, in case something goes wrong as a result of the transition
6. A way to effectively communicate the proposed plan to the entities affected, and incorporate their feedback

We recommend that such criteria be considered when evaluating proposals to transition to new or updated protocols.

5. Security Considerations

This document discusses attributes of protocol transitions. Some types of transition can adversely affect security or privacy. For example, requiring a translator in the middle of the path may hamper end-to-end security and privacy, since it creates an attractive
6. IANA Considerations

This document requires no actions by the IANA.

7. IAB Members at the Time of This Writing

Jari Arkko
Ralph Droms
Ted Hardie
Joe Hildebrand
Russ Housley
Lee Howard
Erik Nordmark
Robert Sparks
Andrew Sullivan
Dave Thaler
Martin Thomson
Brian Trammell
Suzanne Woolf

8. Informative References


Appendix A. Case Studies

Appendix A of [RFC5218] describes a number of case studies that are relevant to this document and highlight various transition problems and strategies (see for instance the Inter-Domain Multicast case study in Section A.4 of [RFC5218]). We now include several additional case studies that focus on transition problems and strategies. Many other equally good case studies could have been included, but, in the interests of brevity, only a sampling is...
explicit here that is sufficient to justify the conclusions in the
body of this document.

A.1. Explicit Congestion Notification

Explicit Congestion Notification (ECN) is a mechanism to replace loss
as the only signal for the detection of congestion, with an explicit
signal sent from a router to the recipient of a packet, then
reflected back to the sender. It was standardized in 2000 in
[RFC3168], and the mechanism consists of two parts: congestion
detection in the IP layer, reusing two bits of the old IP Type of
Service (TOS) field, and congestion feedback in the transport layer.
Feedback in TCP uses two TCP flags, ECN Echo and Congestion Window
Reduced. Together with a suitably configured active queue management
(AQM), ECN can improve TCP performance on congested links.

The deployment of ECN is a case study in failed transition followed
by possible redemption. Initial deployment of ECN in the early and
mid 2000s led to severe problems with some network equipment,
including home router crashes and reboots when packets with ECN IP or
TCP flags was received [TSV2007]. This led to firewalls stripping
ECN IP and TCP flags, or even dropping packets with these flags set.
This stalled deployment. The need for both endpoints (to negotiate
and support ECN) and on-path devices (to mark traffic when congestion
occurs) to cooperate in order to see any benefits from ECN deployment
was a further issue. The deployment of ECN across the Internet had
failed.

In the late 2000s, Linux and Windows servers began defaulting to
"passive ECN support", meaning they would negotiate ECN if asked by
the client, but would not ask to negotiate ECN by default. This
decision was regarded as without risk: only if a client were
explicitly configured to negotiate ECN would any possible
connectivity problems surface. Gradually, this has increased server
support in the Internet from near zero in 2008, to 11% of the top
million Alexa web servers in 2011, to 30% in 2012, to 65% in late
2014. In the meantime, the risk to connectivity of ECN negotiation
has reduced dramatically [PAM2015], leading to ongoing work to make
Windows, Apple iOS, OSX, and Linux clients negotiate ECN by default.
It is hoped that a critical mass of clients and servers negotiating
ECN will provide an incentive to mark congestion on ECN-enabled
traffic, thus breaking the logjam.

A.2. Classless Inter-Domain Routing (CIDR)

TODO To be filled in... [RFC4632]
A.3. Internationalized Domain Names

TODO To be filled in...

A.4. IPv6

Twenty-one years after publication of [RFC1883], the transition to IPv6 is still in progress. The first document to describe a transition plan ([RFC1933], later obsoleted by [RFC2893]) was published less than a year after the protocol itself. It recommended co-existence (dual-stack or tunneling technology) with the expectation that over time, all hosts would have IPv6, and IPv4 could be quietly retired.

In the early stages, deployment was limited to peer-to-peer uses, tunneled over IPv4 networks. For example, Teredo [RFC4380] aligned the cost of fixing the problem with the benefit, and allowed for incremental benefits to those who used it.

Operating System vendors had incentives because with such tunneling protocols, they could get peer-to-peer apps working without depending on any infrastructure changes. That resulted in the main apps using IPv6 being in the peer-to-peer category (BitTorrent, XBox gaming, etc.).

Router vendors had some incentive because IPv6 could be used within an intra-domain network more efficiently than tunneling, once the OS vendors already had IPv6 support and some special-purpose apps existed.

For content providers and ISPs, on the other hand, there was little incentive for deployment: there was no incremental benefit to deploying locally. Since everyone already had IPv4, there was no network effect benefit to deploying IPv6. Even as proponents argued that workarounds to extend the life of IPv4--such as CIDR, NAT, and stingy allocations--made it more complex, IPv4 continued to work well enough for most applications.

Workarounds to NAT problems documented in [RFC6269] and [RFC7021] included ICE, STUN, and TURN, technologies that allowed those experiencing the problems to deploy technologies to resolve them. As with end-to-end IPv6 tunneling (e.g., Teredo), the incentives there aligned the cost of fixing the problem with the benefit, and allowed for incremental benefits to those who used them. The IAB discussed NAT technology proposals [RFC3424] and recommended they be considered short-term fixes, and said that proposals must include an exit plan, such that they would decline over time. In particular, the IAB warned against generalizing NAT solutions, which would lead to
greater dependence on them. In some ways, these solutions, along
with other IPv4 development (e.g., the workarounds above, and
retrofitting IPsec into IPv4) continued to reduce the incentive to
deploy IPv6.

Indeed, not until a few years after IPv4 runout in various Regional
Address Registry (RIR) regions did IPv6 deployment significantly
increase. The RIRs and others conducted surveys of different
industries and industry segments to learn why people did not deploy
IPv6 [IPv6Survey2011] [IPv6Survey2015], which commonly listed lack of
a business case, lack of training, and lack of vendor support as
primary hurdles. Arguably forward-looking companies collaborated
with ISOC on World IPv6 Day and World IPv6 Launch to jump start
global IPv6 deployment, and arguably their work gave vendors
incentives to support IPv6 well. Key aspects of World IPv6 Day and
World IPv6 Launch that contributed to their successes were the
communication mechanism, and the measurement metrics and contingency
plans that were announced in advance.

Several efforts have been made to mitigate the lack of a business
case. Some governments (South Korea, Japan) provided tax incentives
to include IPv6. Other governments (Belgium, Singapore) mandated
IPv6 support by private companies. Few of these had enough value to
drive significant IPv6 deployment.

The concern about lack of training is often a common issue in
transitions. Because IPv4 is so ubiquitous, its use is routine and
simplified with common tools, and it is taught in network training
everywhere. While IPv6 deployment was low, ignorance of it was no
obstacle to being hired as a network administrator or developer.

Organizations with the greatest incentives to deploy IPv6 are those
which continue to grow quickly, even after IPv4 free pool exhaustion.
Thus, ISPs have had varying levels of commitment, based on the growth
of their user base, services being added (especially video over IP),
and the number of IPv4 addresses they had available. Cloud-based
providers, including CDN and hosting companies, have been major
buyers of IPv4 addresses, and several have been strong deployers and
advocates of IPv6.

Different organizations will use different transition models for
their networks, based on their needs. Some are electing to use
IPv6-only hosts in the network with IPv6-IPv4 translation at the
edge. Others are using dual-stack hosts with IPv6-only routers in
the core of the network, and IPv4 tunneled or translated through them
to dual-stack edge routers. Still others are using native dual-stack
throughout the network, but that generally persists as an interim
measure: adoption of two technologies is not the same as
transitioning from one technology to another. Finally, some walled gardens or isolated networks, such as management networks, use IPv6-only end-to-end.

It is impossible to predict with certainty the path IPv6 deployment will have taken when it is complete. Lessons learned so far include aligning costs and benefits (incentive), and ensuring incremental benefit (network effect, or backward compatibility).

A.5. HTTP/2

HTTP/2 [RFC7540] is a new version of the popular HTTP protocol [RFC7230]. The original versions of HTTP (0.9 [HTTP0.9], 1.0 [RFC1945], and 1.1 [RFC2616]) have only small differences; each iteration made small improvements over the previous version without making major changes.

The changes in HTTP/2 are largely aimed at improving performance. The primary improvement is request multiplexing, which is supported by request prioritization and flow control. HTTP/2 includes efficiency improvements with header compression [RFC7541] and binary framing.

A.5.1. Bundling of Features with New Versions

The bundling of additional constraints on a new version of a protocol could affect adoption by making the transition more costly. However, the transition to a new version also represents an opportunity to improve multiple aspects of a protocol at the same time.

The HTTP working group decided that a new version of the protocol represented an opportunity to improve security posture. HTTP/2 is much stricter about its use of TLS. In particular, a long list of TLS cipher suites are prohibited, constraints are placed on the key exchange method, and renegotiation is prohibited. These changes did cause deployment problems. Though most were minor and transitory, disabling renegotiation caused problems for deployments that relied on the feature to authenticate clients and prompted new work to replace the feature.

A number of other features or characteristics of HTTP were identified as potentially undesirable. Several such features were considered for removal during the design process. This included trailers, the 1xx series of responses, certain modes of request forms, and the unsecured (http://) variant of the protocol. For each of these, the risk to the successful deployment of the new version was considered to be too great to justify removing the feature. However, deployment of the unsecured variant of HTTP/2 remains extremely limited.
A.5.2. Planning for Replacement

HTTP/1.1 provides a mechanism, Upgrade, to transition to an entirely different protocol. That same facility was little used other than to enable the use of WebSockets [RFC6455]. However, with performance being a primary motivation for HTTP/2, a new mechanism was needed to avoid spending an additional round trip on this negotiation. A new mechanism was added to TLS to permit the negotiation of the new version of HTTP: Application Layer Protocol Negotiation (ALPN) [RFC7301]. Upgrade was used only for the unsecured variant of the protocol.

ALPN was identified as the way in which future protocol versions would be negotiated. The mechanism was well-tested during development of the specification, which proved that new versions could be deployed safely and easily using ALPN. Several draft versions of the protocol were successfully deployed during protocol development, and version negotiation was never shown to be an issue.

Confidence that new versions would be easy to deploy if necessary lead to a particular design stance that might be considered unusual in light of the advice in RFC 5218 [RFC5218], though is completely consistent with RFC 6709 [RFC6709]: many of the ways in which the protocol might be extended were removed unless an immediate need was understood. This decision was made on the basis that it would be easier to revise the entire protocol than it would be to ensure that an extension point was correctly specified and implemented such that it would be available when needed.

Author’s Address

Dave Thaler (editor)
Microsoft
One Microsoft Way
Redmond, WA  98052
US

Email: dthaler@microsoft.com