May 2023
Geoff Huston

Failed Expectations

In a recent workshop I attended, reflecting on the evolution of the Internet over the past 40 years, one of the takeaways for me is how we've managed to surprise ourselves in both the unanticipated successes we've encountered and in the instances of failure when technology has stubbornly resisted to be deployed despite our confident expectations to the contrary! What have we learned from these lessons of our inability to predict technology outcomes? Are the issues related to the aspects of the technology? Are they embedded in the considerations behind the expectations about how a technology will be adopted? Or do the primary issues reside at a deeper level relating to economic and even political context? Let's look at this question of failed expectations using several specific examples drawn from the last 40 years of the Internet's evolution.

The Public Debut of the Internet (and the demise of OSI)

The 1980's were heady days in the computer and communications realm. The stranglehold of the mainframe computer over the entire computing domain was being challenged by the rise of smaller personal use devices. Moore's Law was starting to exert its influence and year by year computers were getting smaller, faster, and cheaper. The IBM personal computer was launched in 1981, and this landmark product breached the barriers to entry into the consumer market as well as into the enterprise domain. The computer industry had to quickly pivot to respond to unprecedented market demand for these personal use devices. There was an extraordinary pressure to diversify and break out from a central mainframe vendor dominated environment to one of a distributed computing environment that was trying to span across this new generation of edge-based personal computers. The essential glue to hold all this together was a communications network. The proprietary communications protocols of the day were a poor fit to the demands of this novel diverse computing environment and there was a strong interest in vendor-neutral (or "open") network technologies. The rapid rise of Ethernet as the dominant local area network medium illustrated this shift to open technologies as distinct from vendor-based solutions.

In response to this need for a common vendor-neutral communications framework two technologies emerged as potential candidate open communication solutions: the Open System Interconnection framework (OSI) and the Internet Protocol Suite. The exercise of examining the differences in approach between these two technologies from technical and political perspectives is outside of our scope here, but there is one aspect I would like to highlight here. The OSI was deliberately positioned as being the open systems communications protocol for the entire industry. The expectation was that this would be foundation of the digital communications realm, and it was designed with this outcome in mind. The expectation for the OSI protocol suite was success. It may have been a convenient fiction at the time, but the Internet carried with it an entirely different set of professed expectations. The Internet work was claimed at the time to be nothing more than an experiment in packet switching communications technologies both as a learning experience and as a research opportunity. At some point the larger communications industry would take the learnings from this work and apply them to the open communications developmental efforts, or at least that was the intended outcome of this experiment. The professed expectation for the Internet was that it would be superseded at some point.

It was another decade or so, in June 1992, when these expectations were put to the test. Following some dire warnings about the scaling properties of the Internet's routing system and the likely near-term exhaustion of the Class B IPv4 address pool, the Internet Architecture Board (IAB) considered the next steps for the Internet technology that could avert these looming issues. The IAB outlined a direction that would reposition the Internet upon the OSI foundations of the CLNS transport and OSI endpoint addresses. This was in many ways a conservative decision by the IAB at the time, echoing a broader industry sentiment. For example, at that time Digital Equipment (DEC) was a dominant actor in the computer industry, and its proprietary communications protocol, DECnet, was widely used, not only in enterprise and campus communities but also in many wide area network contexts, including the global High Energy Physics Network (HEPnet) and the NASA Science Network. DEC had announced its intention to scale up its network with DECnet Phase V, and this was intended to be based on the OSI protocol suite.

With this IAB action, both the expectations of the OSI effort and the nascent Internet were elevated into prominence. The scaling issues that triggered the IAB consideration of the future paths of the internet were brought about by the rapid adoption of IP outside of its original research base. These early adopters of IP were not looking to the Internet as a short-term experiment, but as a peer member of a collection of communications protocols that were part of the multi-protocol communications world of that period. It was also evident at that time that OSI was in trouble in terms of its further prospects. The efforts by the public sector to prop of the OSI efforts through government OSI profiles (GOSIP) in many countries was ineffectual, and while DEC had committed to OSI in its chosen direction, DEC's own prospects were foundering.

We were in the middle of not only a transition of communications protocols, but also a transition of computer platforms. The locus of investment in computing was no longer the mainframe central computing facility. The environment was a distributed environment and a diverse one as well. The previous tightly coupled relationship between a proprietary operating system with a proprietary communications suite did not readily match this diverse environment. Within the margins of mainframe sales to sustain the continued development of proprietary operating systems, the computing industry turned to Unix as the open alternative. (With of course the notable exception of the personal computer sector, which was rapidly dominated by Microsoft's Windows platform). With this defect decision to embrace the Unix platform came the open implementation of the IP communications protocol. Whether the IP protocol rephrased its own expectations of longevity based on its own merits, or whether it was the shift towards the Unix platform that provided sufficient impetus to the Internet by association is debatable even today, thirty years later. But for me, the point in time in 1992 when the IAB announced a decision for the Internet's direction was the point when these expectations for both IP and OSI were cast aside.

However, we still have some of the legacy of this perversely failed expectation of the IP protocol suite as an experiment. There was no substantial response within the protocol design to the issue of security and robustness in a hostile environment. Many of the shortcomings on this area were appreciated at the time, but the impetus to take the time to craft a more resilient framework simply did not exist. There was little in the way of direct provisions in the protocol relating to security and resilience of protocol operation in a hostile environment. The observation that security was an afterthought in the IP protocol design is true in many ways, but I would argue that this was a deliberate omission at the time. It would be up to IP's successor to tackle such considerations of security and resilience in a more integrated manner.

> It is probably worth also noting that to think, or to go further and assume, that elements of the networked environment could be deliberately perverted to act in ways that were hostile to the interests of the end client users of the network was an unusual and perverse thought at that time. Our model of the telephone network included the consideration that the network operator was a disinterested but commonly trusted common infrastructure operator. For a

network operator to operate their network in ways that undermined such trust was a self-defeating proposition, as it would drive away the network's users.

This concept of the common network infrastructure as a commonly trusted foundational component of the communications environment certainly influenced the design thinking in the formative stages of computer communications networks. It was not until we took the time to reflect on combining the layers of the network's command and control systems (such as routing and addressing) into the same channel as the carried data did we start to realise the extent that these control systems could be perverted as a result.

The Saga of IPv6

With this recasting of expectations for the Internet, the effort in the evolution of the underlying IP protocol suite was coordinated through the IETF. Vendors, corporates and researchers were not following their own directions, and, at least in North America the common effort was loosely coordinated in the IETF.

The picture was not so clear on the other side of the Atlantic Ocean, and the European efforts around OSI and X.25 packet transport, notable by continued support from the telco sector, continued.

The IETF entered a period of re-grouping to consolidate the industry efforts on evolution of the IP protocol framework, discarding both the OSI elements that had previously been considered, and also discarding the up-to-then composition and role of the Internet Architecture Board within the IETF on the way. The IETF of that time reflected a profound optimism about the efficacy and appropriateness of the Internet protocol suite and quickly rephrased its expectations for the Internet around the expectation of broader and longer-term adoption.

It is also worth noting that the internal realignment within the IETF in response to the way in which concerns over imminent IPV4 address exhaustion and scalability of routing were to be addressed did not in fact address these concerns per se. The outcome for the IETF was to take the position that IP itself needed to directly address these scaling issues, and it was up to the IETF to take the lead on this next generation network protocol effort.

This project was undertaken in an unusual context of planning for the consequences of the failure of the current IP protocol suite while this very same protocol was in right in the middle of a period of euphoric success! In the early 1990's the larger computer industry had embraced a diverse approach to computing services, using combinations of personal computers, workstations, minicomputers, and mainframe servers. While it was not the only protocol around at the time, IP was the only capable open protocol that could be used on a license-free basis and implemented based on open-source code. Despite the longer-term issues of routing scaling and IPv4 address exhaustion, the demand for IPv4 was growing at unprecedented levels. It appeared to many at the time that it was due to the essential characteristics of the protocol design, from the lower levels of datagram messages, through to a reliable streaming protocol layered on top of this foundation, and the end-to-end design principle, were the key elements of the runaway success of the IP protocol.

What the IETF learned from this experience was that the unanticipated success of IP was due to the aspects that differentiated it from the OSI protocol suite, namely clear, simple protocol specifications that were openly available and supported interoperable implementations. The mantra of "*rough consensus and running code*" was enthusiastically adopted by the IETF, while at the same time the considerations of careful engineering, conservative design and minimal constraint were considered as secondary considerations. The application of this process appeared to take precedence over the quality of the

product that was generated by this process. Freely available implementations of the protocol helped as well. If this IETF process was what caused IPv4's rapid success, then all the IETF needed to do was to repeat this behaviour in whatever they up with as a scalable successor to IPv4, and then success was assured.

The IETF came up with a successor protocol to IPv4, IPv6, in 1994. It was a conservative choice, in that there was little in the way of substantive change to the IPv4 protocol. The address fields were expanded to 128 bytes in length, the fragmentation control fields became an optional extension header, and a Flow label was added to the header. The service offered by the IP layer to the transport layer was unchanged from IPv4, so that transport protocols could operate unchanged. The expectation was that IPv6, being much the same as IPv4 in terms of functionality, open specification, and openly available implementations, would have a similar adoption momentum at the appropriate time.

There was one consideration in this design effort that was an acknowledged major impediment. IP is not a relay protocol. Network devices, or routers, have a very limited role. The forward packets, but they do not rewrite them. There is no intrinsic ability for a network element to transform IPv4's 32-bit representation of the packet's intended destination to a 128-bit destination. To put it another way. IPv6 could never be backward compatible to IPv4. No matter how similar IPv6 was to IPv4, or how different for that matter, they were still distinct protocols, and to transition the Internet from IPv4 to IPv6 required every network element and every host to be equipped with the capability to support both protocols. Like it or not, the transition entailed a period of operating a multi-lingual network. This is not a failing in IPv6. It's an intrinsic property of IPv4.

This lack of backward compatibility was acknowledged at the time, but it was felt that the lack of any other option to address these address-scale shortcoming in IPv4 meant that the Internet would appreciate these implicit costs of transition and would undertake the transition. After all, the shift from running communications networks as a collection of disparate overlays in a chaotic predominately circuit switched environment to the adoption of IPv4 and packet-switched networks was seen to be a far more traumatic transition! If the industry could do this once, then it could do it again.

However, there were three other measures that were adopted at around that time that had a bearing on these expectations about the adoption of IPv6.

The first of these was a change to our routing protocols to allow the network/host boundary to be variably set, shifting away from the old Class A, B, and C fixed boundary prefix sets. In protocol terms the change was quite simple: a network prefix length (of between 0 and 32 bits) was appended to each address prefix. In terms of the prospects of imminent address exhaustion, the change was profound, in that the address exhaustion issue related specifically to the exhaustion of the pool of Class B addresses.

> The number of addresses allocated to a network was based on a projected demand over an indefinite time into the future, and the class-based structure meant that there were only a few possible allocation sizes (256, 65,536 or 16,777,216 individual addresses). The imminent exhaustion of the address space in IPv4 was not due to the massive explosion in the size of the Internet per se. It was primarily due to the inefficiency of the address allocation framework.

Classless network prefixes allowed the address allocation function to match a network host count more exactly to the network's address prefix. The effectiveness of this change to the address structure was that it could be achieved without changing the behaviour of any end host. In late 1994 there were under 1,000 distinct networks, so once the basic routing protocols were refined to carry classless network prefixes, the process of coordinating deployment was sufficiently small that it was not a challenging

proposition. By mid-1994 the deployment of the classless variant of the inter-domain routing protocol, (BGP 4) was effectively complete and the trajectory of address consumption was changed.

The second measure was the inception of the Regional Internet Registries (RIRs). The previous address allocation framework operated with no upfront or ongoing cost to the address holder. If we were willing to invest time and effort in improving the address utilisation efficiency within the address allocation framework, then the prospect of address exhaustion could be deferred for years, if not decades. The RIRs provided the means to perform this increased investment in the address allocation framework. Address allocation was changed to reflect a transactional nature, where each additional allocation reflected only near-term projections of incremental demand in the network, and costs were imposed on the function, both in upfront and ongoing costs. In the four years from 1990 to 1994 some 600M IPv4 addresses were allocated. In the latter part of that same decade, from 1996 until 2000 this figure dropped to a total of some 250M addresses, while the size of the network in terms of the host count grew 10-fold.

The third measure was one that was largely decried by many in the IETF at the time but was even more effective in terms of deferring address exhaustion. This measure was in fact in two parts. The first was a shift in the Internet's architecture from the peer-to-peer symmetrical design (as in the telephone service mode every connected handset can both make and receive calls) to an asymmetric client/server design where clients can only communicate with servers. This client/server architecture was an extremely close fit to the intermittent connection model of dial-up connectivity used extensively in the early to mid-1990's. The second part is the observation that as clients only initiate transactions then the client only needs an IP address while it is actively engaged in a transaction. IP addresses can be shared. To facilitate this address sharing, network address translation technologies (NATs) were deployed in many access networks. Client networks could be addressed from a common private address pool to allow clients to distinguish themselves from each other in the local network context, and when they initiated a connection to an external server the external gateway, or NAT unit, could borrow an address from an external address pool for the duration and return it to the pool after the transaction is concluded. The host has no need to know that address translation is taking place.

The combination of these three measures has had a profound impact on the projections of address exhaustion. Instead of exhausting the IPv4 address pool by the mid 1990's, the result of adoption of these measures was that the exhaustion date was pushed out to the late 2030's! Even the introduction of the mobile handheld devices and the Internet of Things, both of which have a deployment population of billions of devices have not had a dramatic impact on this picture. Did we jump too early with IPv6? Were we too keen to embark upon a new protocol?

If this entire exercise was to avoid unnecessary network distortions due to address scarcity, then we have failed in this respect. If addresses were not to be regarded as property to be bought and sold, then again, we have failed. I suspect that to a large extent we failed in appreciating the role risk takes in a diverse competitive environment. In such a diverse environment risk may be present in various levels to various entities. If this variation in risk can be exploited for competitive advantage, then it is likely that it will happen. In the case of IPv6, early adopters did not enjoy a competitive advantage and it has been the late adopters who were in an advantaged position by being able to defer the cost of a transition to dual stack.

The DNS and DNSSEC

The design of the DNS represents a number of uneasy compromises. A DNS query (and response) is forerunner of almost every transaction over the Internet. Unsurprisingly, the DNS as a query/response protocol was designed to be fast, lightweight, versatile and simple. The DNS uses UDP, where the client sends a query in a single UDP packet to the server, and the server responds with a response that contains a copy of the query and the server's response. These UDP transactions are not encrypted.

The DNS infrastructure has an internal structure of various servers. Authoritative servers can respond authoritatively for queries that relate to the zone(s) that they are configured to serve. Recursive resolvers are intermediaries that perform the full name resolution function, performing a sequence of discovery

queries that are intended to identify the servers that are authoritative for the DNS domain in the query, and the label query that is directed to these authoritative servers. Recursive resolvers may cache DNS answers and cached entries may be used in response to subsequent queries as appropriate. There are also stub resolvers, which are simple clients that direct all their queries to recursive resolvers.

There are a number of vulnerabilities that come with this approach. If a querier is provided with a deliberately falsified response, then there is no way the querier can tell that this has occurred. Given that the querier has no further information that would contradict the false response it has no choice but to accept this DNS response as genuine. The corrupted response may have been generated as a result of some form of traffic interception within the network, or it may have been caused by a compromised authoritative server or recursive resolver. In the case of responses generated from the resolver's cache, it may be that the cache contents have been corrupted.

The response to this vulnerability was to place a framework of digital signature into the DNS, so that every DNS record in a zone would have a digital signature. If the record has been altered in an unauthorised manner, then the signature would no longer match the record contents and the receiver of the response could detect that the response had been tampered with. The intent was to make the addition of these digital signatures to be fully compatible with the DNS, so the signature is treated as another resource record in the zone file. If the query includes a flag to indicate that the query wants to be additionally provided with the digital signature of the response, it will set a flag in the query. There are two additional aspects of this design to complete the framework. The first is the way in which non-existent resource record types and non-existent labels are handled. The approach used by DNSSEC is the definition of a new resource record type, the NSEC record, attached to each distinct label in the zone. This record, which has its own digital signature, enumerates the resource types that are defined for this label, and also the next label in the zone, accordingly to a sorted enumeration of all labels defined in the zone. Negative responses carry this NSEC record and its digital signature as "proof" on nonexistence. The second aspect is the issue of authenticity of the key used to generate these digital signatures. This is addressed by having the zone's parent publish a signed record in its zone which is the hash of the public key.

The specification of DNSSEC has taken many years. The earliest version of this technology in the RFC document series is RFC 2065, published in January 1997. The DNSSEC specifications have had several subsequent updates, reflecting an evolving understanding of the operational requirements of this form of securing DNS record content. Study continues in the areas of automated signalling between child and parent for key changes and in changes to some negative record definitions in the light of the increasing use of dynamic signers.

DNSSEC could be seen as a success for the IETF process. A vulnerability that was being exploited was identified, a solution to that vulnerability was defined and deployment has proceeded.

But to call this a success is premature. It's not quite the case that nobody signs and nobody validates. But it's not exactly a runaway success either. Some measurement would help here.

To measure the extent to which validation is performed we can use a *beaconing* technique. Here a resource name is published in the DNS, but its DNSSEC signature is deliberately corrupted, so it will fall all DNSSEC validation attempts. We then enrol a diverse (and hopefully representative) sample set and measure the proportion of tests that fail DNS resolution of this name. In our measurements at APNIC we've found that the validation rate is some 30% of the Internet's user base and has been at this level of the past 14 months or so (https://stats.labs.apnic.net/dnssec/XA?hc=XA&hx=0&hv=1&hp=1&hr=1&w=1&p=0&r=0). Even this measurement result is misleading, in so far is it's important to understand where DNSSEC validation occurs. It appears that that very few stub DNS resolvers perform independent validation of signed DNS responses. The stub resolver simply believes what it is told by its recursive resolver, and if the DNS response from the recursive resolver were to be altered in any way along the path to the stub resolver, then the stub resolver would be unable to detect such tampering. While only 30% of users sit

behind DNSSEC-validating resolvers, a (far?) smaller number of these users use DNSSEC validation in their local stub resolver.

Measuring the extent to which DNS data is signed is harder to measure because of the indeterminate nature of the DNS name space. We have found it infeasible to perform a comprehensive walk-through of the DNS name space and the various efforts to estimate a so-called "signing rate" appear to sit within the range of 5% - 10% of zones. It's still a stubbornly small number.

Let's get back to expectations here. The IETF's expectation was that a standard specification of a technology that would permit clients to assure themselves of authenticity and currency of DNS responses would be in the interests of both DNS name publishers and DNS clients. Not only would such a measure mitigate the risk from a number of known attacks on the DNS, but it would facilitate using the DNS beyond a single name-to-address mapping, allowing other name-related attributes to be loaded into the DNS in a reliable manner. It seemed impossible to envisage a scenario where both name holders and users of names would shun the use of DNSSEC.

And yet that's what has happened.

Why?

DNSSEC can add additional queries to the process of name resolution. A DNSSEC-aware client normally requests that DNSSEC digital signatures be added to the response, and this additional data has the potential to trigger large response behaviours in the DNS. The response may be fragmented in UDP, which then has its issues with increased probability of packet loss, or it may trigger DNS buffer size limits, causing the query to be repeated over TCP. Additionally, the DNSSEC-validating resolver has to query each of the parent zones for DS and DNSKEY records to authenticate the zone's signing key. These additional queries also take time and may also trigger large response behaviours.

This is a case where the theory sounds far worse than practice. DNS caching is highly effective in eliminating extraneous query behaviour. However, when looking at technology adoption factors it’s perceptions that count a whole lot more than measurements, and if the perception is that DNSSEC-signed zones slow down the DNS name resolution process then this perception of compromised DNS performance is an extremely important disincentive for adoption.

There is a second factor which also appear to act as a powerful deterrent for adoption of DNSSEC, which is increased fragility of the DNS. DNSSEC key management is not straightforward, and while it's easy to make it more complex by adding more zone publishers, dynamic signing and frequent key transitions, it's challenging to make it simpler and more robust. There is a strong aversion to inadvertently publish a broken DNSSEC zone that would cause these validating resolvers to make the zone as unresolvable.

So, we have a situation of some clear costs in terms of resolution speed and robustness of zone publication and still little in the way of identified benefit.

Surely having a DNS response that a user could rely upon is a benefit? Oddly enough, it’s not necessarily so. The issue here is that the overall majority of names are not DNSSEC signed, and evidently the overall majority of DNS name resolution operations resolve names that are not DNSSEC-signed. In the majority of cases an application cannot count on any assurances that DNSSEC signing and validation may provide. The application is left to fend for itself. Even if the DNS name resolution was verified locally, the end user application still has to factor in the potential for routing hijacks to direct the user to an incorrect service endpoint. The conclusion is that the application should properly assume that any and all connections are untrusted, and it has to perform its own validation to establish the bona fides of the application server in any case. For example, a browser will still insist on using a TLS connection with validation of the claimed identity of the server with a WEBPKI validation operation irrespective of

whether the server's IP address was discovered using local DNSSEC validation of the DNS information or not.

So, while the benefits of DNSSEC signing and validation sound compelling, they have been insufficiently compelling to motivate widespread adoption.

This failure of the expectations for DNSSEC adoption appears to be a perverse form of failure, as the DNSSEC approach tried to meet many of the technology adoption prerequisites. The technology was largely backward compatible, in that additional records used by DNSSEC were conventional DNS records. A non-DNSSEC-aware resolver can function perfectly normally when querying a DNSSEC-signed zone, as can a DNSSEC-aware resolver operating with an unsigned zone. It is a requisite condition that a parent must be DNSSEC-signed before the child zone can be signed, but within a parent zone each delegated child zone can decide whether to DNSSEC-sign the zone independently of the signing status of the other zones. DNSSEC has avoided extraneous complexity by eschewing X.509 public key certificates, and instead has used a far simpler approach that is based around the keys themselves. It is designed to achieve a simple single objective, namely, to detect content alteration. It does not attempt to broaden its objectives in the area of secured transport functions, server authentication or similar.

Why then is DNSSEC adoption failing to meet our expectations? I suspect that the lengthy gestation of DNSSEC has exposed it to the latecomer syndrome. Applications have already had to come to terms with a DNS that cannot be trusted and have had to adopt measures to protect the integrity of network transitions without relying on the integrity of the DNS. Given that applications have addressed this issue to what appears to be their own satisfaction, any additional assurance that DNSSEC can offer is of marginal benefit, and DNSSEC adoption apparently only merits marginal attention.

Successes

In the examples so far, I have selected instances where the expectations as to the adoption and future prospects of the technology were not fulfilled, and looked at the aspects of technology that were relevant to these failed expectations. Of course, there have been many cases where the initial expectations were met, and I'd like to touch upon a few such examples here.

The most outstanding example for me is the TCP protocol. TCP represented a different way of thinking about transport protocol behaviour. To illustrate this, it's useful to compare TCP's behaviour with that of the DECnet protocol. DECnet used a router-to-router protocol (DDCMP) that operated as a reliable flow-controlled protocol. Each router would retain a copy of an outgoing packet until the next hop router had positively acknowledged its receipt. This hop-by-hop control tends to exert pushback to congestion by not acknowledging packets when the local queues fill. This, in turn, exerts pushback on the upstream sending router and so forth back to the sending source. While this mechanism creates a smooth flow response from the network, it is not highly efficient. TCP replaced this hop-by-hop flow control with a single end-to-end flow control. The result of this change was expected to be highly efficient, as indeed it turned out to be once Dave Borman and Van Jacobsen had resolved some outstanding tuning issues. The expectation that we could use the same TCP engine and use it to fill a 10Mbps Ethernet connection, fill a 64Kbps satellite circuit and also load a high capacity 45Mbps trans-continental connection. These days TCP can operate at speeds from as little as a few bits per second to as fast as 10's and even 100's of gigabits per second. It does not end there. We expected this same transport protocol to be efficiently fair, such that when multiple TCP sessions were running across the same network path, they were expected to equilibrate their demands and share the available network resource fairly between the TCP sessions. And TCP has by and large delivered against these expectations.

Why has TCP been able to achieve all this? There are a few aspects of its design that have endured. The major aspect is that of end-to-end control. What regulates the sending end is the stream of ACK packets coming from the receiving end. It also supports a very minimal set of functions. TCP uses a model of an internally unstructured data stream, and the sender's write() calls to pass data from the application to the network are not necessarily matched by corresponding read() calls. It is left to the application to perform

record marking if such internal structuring of data is required by the application. TCP flow control is the responsibility of the sender, not the receiver. This allows TCP servers to innovate in the flow control algorithms that they use while with no change to the TCP client code.

I would also place the BGP routing protocol as a case of an outstanding example of succeeding in accordance with expectations. The set of requirements for an inter domain routing protocol for the Internet are certainly demanding. It must operate in a completely decentralised model. It must scale. We would like the operation of the protocol to consume minimal network capacity, and we would like it to react to changes in network topology in a small number of seconds. Local network operators would like to apply local network policies to locally processed routes without creating any form of external dependency. Autonomous networks need to remain autonomous. And BGP has delivered on these requirements. So far, we are using it to carry more than a million routes across a hundred thousand networks in support of a connected Internet. It is a strong testament to focus in design. BGP addresses just one function in networking, creating a collection of consistent local packet forwarding decisions that collectively result in packets being delivered to their intended destination.

What have we learned?

I must admit to a certain level of cynicism, in that most of the learnings about what makes a successful technology are learnings that the folk working in the Internet were aware on in the late 1980's. At that time the IETF was trying to define its presence by distinguishing itself in the area of technology standards. It's approach was to emphasise that the IETF specifications were based on technology that was deployed, rather than abstract descriptions of some desired functionality (or *paperware about vaporware,* as some in the IETF described the OSI specifications).

The general principles of what made IETF specifications useful could be drawn from a simple list:

- Design for the general case, not a specific scenario.
- Focus on a single function.
- Keep it simple.
- Fulfil a known need.
- Avoid inter-dependencies on third parties wherever possible.
- Backward compatibility and tolerance for piecemeal deployment are essential.

I don't think much has changed over the past thirty or so years, and these same general principles are still relevant today. However there have been a couple of painful lessons in this period which we are coming to terms with.

- Every needless exposure of data can and will be used against the user!

This has been a tough lesson, and I'm not sure that we have fully come to terms with it. Past of this lesson reflects a desire to protect the user from surveillance operations. The Snowden papers were a rude awakening for many IETF participants, where the trail of data generated by open protocols was exhaustively analysed by state entities within the US. The reaction was to assume that all networks were hostile environments, and the objective was to wrap all network transactions with a cloak of encryption.

- Adopters need to be able to realise benefits of adoption from the outset – if benefits are realizable only when everyone deploys, then adoption will stall.

This also has been a tough lesson to learn. The IETF has been all too willing to define specifications where deployment depends on *enlightened common interest*, namely where individual entities may act in ways that are contrary to their immediate self interest in order to further the cause of a common interest. Many security-based systems have this property, where only valid information can be clearly marked as such. Partial adoption where only a subset of information is so marked leaves the relying party in a quandary when presented with unmarked information.

This touches upon a more significant and longer-term issue for the communications enterprise. The sector has been operated by state-sanctioned monopolies of various forms for almost all of the twentieth century. Monopolies have far easier time in dealing with technology-based evolution in that they act as the orchestrator of the diverse supply chains that are used to generate the service offering. They have control over the timing and features that are integrated with the service by virtue of their monopoly position. However, the Internet was a product of the deregulation of the telecommunications industry. This dismembering of the core monopoly that was at the centre of each national enterprise ceded control of the overall orchestration of service providers and technology components to the actions of a market. Market forces replace all other forms of technology acceptance, and adoption is driven by individual entities who are motivated primarily by commercial self-interest. In some cases, the adoption of technology is extremely rapid, such as the shift to the use of Content Distribution Networks and their function of replicated service delivery in place of peering and transit arrangements. In other cases, and the deployment of IPv6 is a good example, the transition is protracted and at times chaotic.

In cases where our expectations about a novel technology or service appear to fail, the problem may well lie in an inability to understand how a market-based economy will accommodate the change. The parameters of market acceptance lie more in the realm of early adopter competitive advantage in preference to common benefit once general adoption has occurred. DNSSEC's struggles with adoption may well be based on this perception, where the incremental benefits of signing a zone are relatively minor, so early adopters incur cost without any major competitive advantage. The enduring value of DNSSEC lie in the ability to place trusted data into the DNS, such as domain name keys, but this value is only realizable if every zone is signed, and every client validates.

Like the early days of the Internet, when it characterised itself as an experiment in the potential of packet-switched communications, we are in the middle of a grander experiment in the use of market forces to shape the direction of the entire communications enterprise. As usual, the results are mixed. The Internet is truly astonishing in its capability, its ubiquity, its affordability, and its utility. In almost every dimension the Internet of 2023 is unimaginable from a stance of 1983, just forty years ago. We are rolling out gigabit access networks, we have launched a network with truly global network using spacecraft orbiting just 550 km above our heads. We have honed and tuned the technology such that the entire mass of human knowledge is accessible to inform a search query in just milliseconds. Computing power exists in a device I wear on my wrist. All of this is revolutionary in terms of the world of 1983, and if it was deregulation that has propelled us to this point then it's a tough case to argue that this has failed us. Yet there is a case to be made that not everything in today's world has been built to our benefit, collectively or individually. We have enlarged markets from national to global in scope and bred corporate behemoths to dominate this global marketplace. These digital monstrosities are equally astonishing in scale of their projection of market power, their overbearing presence such that national interests are no longer able to come to acceptable terms with them, and their cynically exploitative agendas. If there is any vestige of competition left here it's no longer a competition between these digital giants, or even a competition between these private entities and the various state-based national regimes. The remaining competition that we are seeing now if that of a competition with the future, and the efforts of these entities to define an assured role over the coming decades.

The expectation of deregulation of the telco industry was to dismember the stultifying position of the national monopoly incumbents, and to refocus the communications industry to serve the evolving needs of the end user as efficiently and as capably as possible. Yes, it was meant to produce cheaper phone calls, but this form of deregulation was not intended to support cheaper faxes. It was intended to foster an evolution of services such that digital communications that was no longer subservient to the needs of a real time human voice service. And it's clear that in the past 40 years we've come a long way in achieving that. The failure of the expectations of deregulation is that we've successfully replaced a set of national monopolies with a small clique of global behemoths whose exploitative capabilities appear to be far more menacing to the long-term health of our societies than the excesses of the national telcos that they have usurped.

Disclaimer

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

Author

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

www.potaroo.net