June 2018
Geoff Huston

Another 10 Years Later

Ten years ago, I wrote an article that looked back on the developments within the Internet over the period from 1998 to 2008 (https://bit.ly/2yxs7RT). Well another ten years has gone by, and it's a good opportunity to take a little time once more to muse over what's new, what's old and what's been forgotten in another decade of the Internet's evolution.

The evolutionary path of any technology can often take strange and unanticipated turns and twists. At some points simplicity and minimalism can be replaced by complexity and ornamentation, while at other times a dramatic cut-through exposes the core concepts of the technology and removes layers of superfluous additions. The evolution of the Internet appears to be no exception and contains these same forms of unanticipated turns and twists. In thinking about the technology of the Internet over the last ten years, it appears that it's been a very mixed story about what's changed and what's stayed the same.

A lot of the Internet today looks much the same as the Internet of a decade ago. Much of the Internet's infrastructure has stubbornly resisted various efforts to engender change. We are still in the middle of the process to transition the Internet to IPv6, which was the case a decade ago. We are still trying to improve the resilience of the Internet to various attack vectors, which was the case a decade ago. We are still grappling with various efforts to provide defined quality of service in the network, which was the case a decade ago. It seems that the rapid pace of technical change in the 1990's and early 2000's has simply run out of momentum and it seems that the dominant activity on the Internet over the past decade was consolidation rather than continued technical evolution. Perhaps this increased resistance to change is because as the size of the network increases, its inertial mass also increases. We used to quote Metcalf's Law to each other, reciting the mantra that the value of a network increases in proportion to the square of the number of users. A related observation appears to be that a network's inherent resistance to change, or inertial mass, is also directly related to the square of the number of users as well. Perhaps as a general observation, all large loosely coupled distributed systems are strongly resistant to efforts to orchestrate a coordinated change. At best, these systems respond to various forms of market pressures, but as the Internet's overall system is so large and so diverse these market pressures manifest themselves in different ways in different parts of this network. Individual actors operate under no centrally orchestrated set of instructions or constraints. Where change occurs, it is because some sufficiently large body of individual actors see opportunity in undertaking the change or perceive unacceptable risk in not changing. The result for the Internet appears to be that some changes are very challenging, while others look like natural and inevitable progressive steps.

But the other side of the story is one that is about as diametrically opposed as its possible to paint. Over the last decade we've seen another profound revolution in the Internet as it embraced a combination of wireless-based infrastructure and a rich set of services at a speed which has been unprecedented. We've seen a revolution in content and content provision that has not only changed the Internet, but as collateral damage the Internet appears to be decimating the traditional newspaper and broadcast television sectors. Social media has all but replaced the social role of the telephone and the practice of letter writing. We've seen the rise of the resurgence of a novel twist to the old central mainframe service in the guise of the 'cloud' and the repurposing of Internet devices to support views of a common cloud-hosted content that in many ways mimic the function of display

terminals of a bygone past. All of these are fundamental changes to the Internet and all of these have occurred in the last decade!

That's a significant breadth of material to cover, so I'll keep the story to the larger themes, and to structure this story, rather than offer a set of unordered observations about the various changes and developments over the past decade, I'll use a standard model of a protocol stack as the guiding template. I'll start with the underlying transmission media and then looking at IP, the transport layer, then applications and services, and closing with a look at the business of the Internet to highlight the last decade's developments.

Below the IP Layer

What's changed in network media?

Optical systems have undergone sustained change in the past decade. A little over a decade ago production optical systems used simple on-off keying to encode the signal into the optical channel. The speed increases in this generation of optical systems relied on improvements in the silicon control systems and the laser driver chips. The introduction of wavelength division multiplexing in the late 1990's allowed the carriers to greatly increase the carrying capacity of their optical cable infrastructure. The last decade has seen the evolution of optical systems into areas of polarisation and phase modulation to effectively lift the number of bits of signal per baud. These days 100Gbps optical channels are commonly supportable, and we are looking at further refinements in signal detection to lift that beyond 200Gbps. We anticipate 400Gbps systems in the near future, using various combinations of a faster basic baud rate and higher levels of phase amplitude modulation, and dare to think that 1Tbps is now a distinct near term optical service.

Radio systems have seen a similar evolution in overall capacity. Basic improvements in signal processing, analogous to the changes in optical systems, has allowed the use of phase modulation to lift the data rate of the radio bearer. The use of MIMO technology, coupled with the use of higher carrier frequencies has allowed the mobile data service to support carriage services of up to 100Mbps in today's 4G networks. The push to even higher frequencies promises speeds of up to 1Gbps for mobile systems in the near future with the deployment of 5G technology.

While optical speeds are increasing, ethernet packet framing still persists in transmission systems long after the original rationale for the packet format died along with that bright yellow coaxial cable! Oddly enough, the Ethernet-defined minimum and maximum packet sizes of 64 and 1500 octets still persist. The inevitable result of faster transmission speeds with constant packet sizes results in an upper bound of the number of packets per second increasing more 100-fold over the past decade, in line with the increase of deployed transmission speeds from 2.5Gbps to 400 Gbps. As a consequence, higher packet processing rates are being demanded from silicon-based switches. But one really important scaling factor has not changed for the past decade, namely the clock speed of processors and the cycle time of memory, which has not moved at all. The response so far has been in increasing reliance of parallelism in high speed digital switching applications, and these days multi-core processors and highly parallel memory systems are used to achieve performance that would be impossible in a single threaded processing model.

In 2018 it appears that we are close to achieving 1Tbps optical systems and up to 20Gbps in radio systems. Just how far and how quickly these transmission models can be pushed into supporting ever higher channel speeds is an open question.

The IP Layer

The most notable aspect of the network that appears to stubbornly resist all forms of pressure over the last decade, including some harsh realities of acute scarcity, is the observation that we are still running what is essentially an IPv4 Internet.

Over this past decade we have exhausted our pools of remaining IPv4 addresses, and in most parts of the world the IPv4 Internet is running on some form of empty. We had never suspected that the Internet would confront the exhaustion of one its most fundamental pillars, the basic function of uniquely addressing connected devices, and apparently shrug it off and continue on blithely. But, unexpectedly, that's exactly what's happened.

Today we estimate that some 3.4 billion people are regular users of the Internet, and there are some 20 billion devices connected to it. We have achieved this using some 3 billion unique IPv4 addresses. Nobody thought that we could achieve this astonishing feat, yet it has happened with almost no fanfare.

Back in the 1900's we had thought that the prospect of address exhaustion would propel the Internet to use IPv6. This was the successor IP protocol that comes with a four-fold increase in the bit width of IP addresses. By increasing the IP address pool to some esoterically large number of unique addresses (340 undecillion addresses, or 3.4×10^{38}) we would never have to confront network address exhaustion again. But this was not going to be an easy transition. There is no backward compatibility in this protocol transition, so everything has to change. Every device, every router and even every application needs to change to support IPv6. Rather than perform comprehensive protocol surgery on the Internet and change every part of the infrastructure to support IPv6, we changed the basic architecture of the Internet instead. Oddly enough, it looks like this was the cheaper option!

Through the almost ubiquitous deployment of Network Address Translators (NATs) at the edges of the network, we've transformed the network from a peer-to-peer network into a client/server network. In today's client/server Internet clients can talk to servers, and servers can talk back to these connected clients, but that's it. Clients cannot talk directly to other clients, and servers need to wait for the client to initiate a conversation in order to talk to a client. Clients 'borrow' an endpoint address when they are talking to a server and release this address for use by other clients when they are idle. After all, endpoint addresses are only useful to clients in order to talk to servers. The result is that we've managed to cram some 20 billion devices into an Internet that only has deployed just 3 billion public address slots. We've achieved this though embracing what could be described as time-sharing of IP addresses.

All well and good, but what about IPv6? Do we still need it? If so, then then are we going to complete this protracted transition? Ten years later the answer to these questions remain unclear. On the positive side, there is a lot more IPv6 around now than there was ten years ago. Service Providers are deploying much IPv6 today than was the case in 2008. When IPv6 is deployed within a Service Provider's network we see an immediate uptake from these IPv6-equipped devices. In 2018 it appears that one fifth of the Internet's users (that itself is now estimated to number around one half of the planet's human population) are capable of using the Internet over IPv6, and most of this has happened in the past 10 years. However, on the negative side the question must be asked: What's happening with IPv6 for the other four fifths of the Internet? Some ISPs have been heard to make the case that they would prefer to spend their finite operating budgets on other areas that improve their customers' experience such as increasing network capacity, removing data caps, acquiring more on-net content. Such ISPs continue to see deployment of IPv6 as a deferable measure.

It seems that today we are still seeing a mixed picture for IPv6. Some service providers simply see no way around their particular predicament of IPv4 address scarcity and these providers see IPv6 as a necessary decision to further expand their network. Other providers are willing to defer the question to some undefined point in the future.

Routing

While we are looking at what's largely unchanged over the past decade we need to mention the routing system. Despite dire predictions of the imminent scaling death of the Border Gateway Protocol (BGP) ten years ago, BGP has steadfastly continued to route the entire Internet. Yes, BGP is as insecure as ever, and yes, a continual stream of fat finger foul-ups and less common but more concerning malicious route hijacks continue to plague our routing system, but the routing technologies in use in 2008 are the same as we use in today's Internet.

The size of the IPv4 routing table has tripled in the past ten years, growing from 250,000 entries in 2008 to slightly more than 750,000 entries today. The IPv6 routing story is more dramatic, growing from 1,100 entries to 52,000 entries. Yet BGP just quietly continues to work efficiently and effectively. Who would've thought that a protocol that was originally designed to cope with a few thousand routes announced by a few hundred networks could still function effectively across a routing space approaching a million routing entries and a hundred thousand networks!

In the same vein, we have not made any major change to the operation of our interior routing protocols. Larger networks still use either OPSF or ISIS depending on their circumstances, while smaller networks may opt for some distance vector protocol like RIPv2 or even EIGRP. The work in the IETF on more recent routing protocols LISP and BABEL seem lack any real traction with the Internet at large, and while they both have interesting properties in routing management, neither have a sufficient level of perceived benefit to overcome the considerable inertia of conventional network design and operation. Again, this looks like another instance where inertial mass is exerting its influence to resist change in the network.

Network Operations

Speaking of network operation, we are seeing some stirrings of change, but it appears to be a rather conservative area, and adoption of new network management tools and practices takes time.

The Internet converged on using the Simple Network Management Protocol (SNMP) a quarter of a century ago, and despite its security weaknesses, its inefficiency, its incredibly irritating use of ASN.1, and its use in sustaining some forms of DDOS attacks, it still enjoys widespread use. But SNMP is only a network monitoring protocol, not a network configuration protocol, as anyone who has attempted to use SNMP write operations can attest.

The more recent Netconf and YANG efforts are attempting to pull this area of configuration management into something a little more usable than expect scripts driving CLI interfaces on switches. At the same time, we are seeing orchestration tools such as Ansible, Chef, NAPALM and SALT enter the network operations space, permitting the orchestration of management tasks over thousands of individual components. These network operations management tools are welcome steps forward to improve the state of automated network management, but it's still far short of a desirable endpoint.

In the same time period as we appear to have advanced the state of automated control systems to achieve the driverless autonomous car, the task of fully automated network management appears to have fallen way short of the desired endpoint. Surely it must be feasible to feed an adaptive autonomous control system with the network's infrastructure and available resources, and allow the control system to monitor the network and modify the operating parameters of network components to continuously meet the network's service level objectives? Where's the driverless car for driving networks? Maybe the next ten years might get us there.

The Mobile Internet

Before we move up a layer in the Internet protocol model and look at the evolution of the end-to-end transport layer, we probably need to talk about the evolution of the devices that connect to the Internet.

For many years the Internet was the domain of the desktop personal computer, with laptop devices serving the needs to those with a desire for a more portable device. At the time the phone was still just a phone, and their early forays into the data world were unimpressive.

Apple's iPhone, released in 2007, was a revolutionary device. Boasting a vibrant colour touch sensitive screen, just four keys, a fully functional operating system, with WiFi and cellular radio interfaces, and a capable processor and memory, it's entry into the consumer market space was perhaps the major event of the decade. Apple's early lead was rapidly emulated by Windows and Nokia with their own offerings. Google's position

was more as an active disruptor, using an open licensing framework for the Android platform and its associated application ecosystem to empower a collection of handset assemblers. Android is used by Samsung, LG, HTC, Huawei, Sony, and Google to name a few. These days almost 80% of the mobile platforms use Android, and some 17% use Apple's iOS.

For the human Internet the mobile market is now the Internet-defining market in terms of revenue. There is little in terms of margin or opportunity in the wired network these days, and even the declining margins of these mobile data environments represent a vague glimmer of hope for the one dominant access provider industry.

Essentially, the public Internet is now a platform of apps on mobile devices.

End to End Transport Layer

It's time to move up a level in the protocol stack and look at end-to-end transport protocols and changes that have occurred in the past decade.

End-to-end transport was the revolutionary aspect of the Internet, and the TCP protocol was at the heart of this change. Many other transport protocols require the lower levels of the network protocol stack to present a reliable stream interface to the transport protocol. It was up to the network to create this reliability, performing data integrity checks and data flow control, and repairing data loss within the network as it occurred. TCP dispensed with all of that, and simply assumed an unreliable datagram transport service from the network and pushed to the transport protocol the responsibility for data integrity and flow control.

In the world of TCP not much appears to have changed in the past decade. We've seen some further small refinements in the details of TCP's controlled rate increase and rapid rate decrease, but nothing that shifts the basic behaviours this protocol. TCP tends to use packet loss as the signal of congestion and oscillates its flow rate between some lower rate and this loss-triggering rate.

Or at least that was the case until quite recently. The situation is poised to change, and change in a very fundamental way, with the debut of Google's offerings of BBR and QUIC.

The Bottleneck Bandwidth and Round-trip time control algorithm, or BBR, is a variant of the TCP flow control protocol that operates in a very different mode from other TCP protocols. BBR attempts to maintain a flow rate that sits exactly at the delay bandwidth product of the end-to-end path between sender and receiver. In so doing, tries to avoid the accumulation of data buffering in the network (when the sending rate exceeds the path capacity), and also tries to avoid leaving idle time in the network (where the sending rate is less than the path capacity). The side effect is that BBR tries to avoid the collapse of network buffering when congestion-based loss occurs. BBR achieves significant efficiencies from both wired and wireless network transmission systems.

The second recent offering from Google also represents a significant shift in the way we use transport protocols. The QUIC protocol looks like a UDP protocol, and from the network's perspective it is simply a UDP packet stream. But in this case looks are deceiving. The inner payload of these UDP packets contain a more conventional TCP flow control structure and a TCP stream payload. However, QUIC encrypts its UDP payload so the entire inner TCP control is completely hidden from the network. The ossification of the Internet's transport is due in no small part to the intrusive role of network middleware that is used to discarding packets that it does not recognise. Approaches such as QUIC allow applications to break out of this regime and restore end-to-end flow management as an end-to-end function without any form of network middleware inspection or manipulation. I'd call this development as perhaps the most significant evolutionary step in transport protocols over the entire decade.

The Application Layer

Let's keep on moving up the protocol stack and look at the Internet from the perspective of the applications and services that operate across the network.

Privacy and Encryption

As we noted in looking at developments in end-to-end transport protocols, encryption of the QUIC payload is not just to keep network middleware from meddling with the TCP control state, although it does achieve that very successfully. The encryption applies to the entire payload, and it points to another major development in the past decade. We are now wary of the extent to which various forms of network-based mechanisms are used to eavesdrop on users and services. The documents released by Edward Snowden in 2013 portrayed a very active US Government surveillance program that used widespread traffic interception sources to construct profiles of user behaviour and by inference profiles of individual users. In many ways this effort to assemble such profiles is not much different to what advertising-funded services such as Google and Facebook have been (more or less) openly doing for years, but perhaps the essential difference is that of knowledge and implied consent. In the advertisers' case this information is intended to increase the profile accuracy and hence increase the value of the user to the potential advertiser. The motivations of government agencies are more open to various forms of interpretation, and not all such interpretations are benign.

One technical response to the implications of this leaked material has been an overt push to embrace end-to-end encryption in all parts of the network. The corollary has been an effort to allow robust encryption to be generally accessible to all, and not just a luxury feature available only to those who can afford to pay a premium. The Let's Encrypt initiative has been incredibly successful in publishing X.509 domain name certificates that free of cost, and the result is that all network service operators, irrespective of their size or relative wealth, can afford to use encrypted sessions, in the form of TLS, for their web servers.

The push to hide user traffic from the network and network-based eavesdroppers extends far beyond QUIC and TLS session protocols. The Domain Name System is also a rich source of information about what users are doing, as well as being used in many places to enforce content restrictions. There have been recent moves to try and clean up the overly chatty nature of the DNS, using query name minimisation to prevent unnecessary data leaks, and the development of both DNS over TLS and DNS over HTTPS to secure the network path between a stub resolver and its recursive server. This is very much a work in progress effort at present, and it will take some time to see if the results of this work will be widely adopted in the DNS environment.

We are now operating our applications in an environment of heightened paranoia. Applications do not necessarily trust the platform on which they are running, and we are seeing efforts from the applications to hide their activity from the underlying platform. Applications do not trust the network, and we are seeing increased use of end-to-end encryption to hide their activity from network eaves droppers. The use of identity credentials within the encrypted session establishment also acts to limit the vulnerability of application clients to be misdirected to masquerading servers.

The Rise and Rise of Content

Moving further up the protocol stack to the environment of content and applications we have also seen some revolutionary changes over the past decade.

For a small period of time the Internet's content and carriage activities existed in largely separate business domains, tied by mutual interdependence. The task of carriage was to carry users to content, which implied that carriage was essential to content. But at the same time a client/server Internet bereft of servers is useless, so content is essential to carriage. In a world of re-emerging corporate behemoths, such mutual interdependence is unsettling, both to the actors directly involved, and to the larger public interest.

The content industry is largely the more lucrative of these two and enjoys far less in the way of regulatory constraint. There is no concept of any universal service obligation, or even any effective form of price control

in the services they offer. Many content service providers use internal cross funding that allows them to offer free services to the public, as in free email, free content hosting, free storage, and similar, and fund these services through a second, more occluded, transaction that essentially sells the user's consumer profile to the highest bidding advertiser. All this happens outside of any significant regulatory constraint which has given the content services industry both considerable wealth and considerable commercial latitude.

It should be no surprise that this industry is now using its capability and capital to eliminate its former dependence on the carriage sector. We are now seeing the rapid rise of the content data network (CDN) model, where instead of an Internet carrying the user to a diverse set of content stores, the content stores are opening local content outlets right next to the user. As all forms of digital services move into CDN hostels, and as the CDN opens outlets that are positioned immediately adjacent to pools of economically valuable consumers, then where does that leave the traditional carriage role in the Internet? The outlook for the public carriage providers is not looking all that rosy given this increasing marginalisation of carriage in the larger content economy.

Within these CDNs we've also seen the rise of a new service model enter the Internet in the form of cloud services. Our computers are no longer self-contained systems with processing and compute resources but look more and more like a window that sees the data stored on a common server. Cloud services are very similar, where the local device is effectively a local cache of a larger backing store. In a world where users may have multiple devices this model makes persuasive sense, as the view to the common backing store is constant irrespective of which device is being used to access the data. These cloud services also make data sharing and collaborative work far easier to support. Rather than creating a set of copies of the original document and then attempt to stitch back all the individual edits into a single common whole, the cloud model shares a document by simply altering the document's access permissions. There is only ever one copy of the document, and all edits and comments on the document are available to all.

The Evolution of Cyber Attacks

At the same time as we have seen announcements of ever increasing network capacity within the Internet we've seen a parallel set of announcements that note new records in the aggregate capacity of Denial of Service attacks. The current peak volume is an attack of some 1.7Tbps of malicious traffic.

Attacks are now commonplace. Many of them are brutally simple, relying on a tragically large pool of potential zombie devices that are readily subverted and co-opted to assist in attacks. The attacks are often simple forms of attack, such as UDP reflection attacks where a simple UDP query generates a large response. The source address of the query is forged to be the address of the intended attack victim, and not much more need be done. A small query stream can result in a massive attack. UDP protocols such as SNMP, NTP, the DNS and memcache have been used in the past and doubtless will be used again.

Why can't we fix this? We've been trying for decades, and we just can't seem to get ahead of the attacks. Advice to network operators to prevent the leakage of packets with forged source addresses, RFC 2827, was published in two decades ago in 1998. Yet massive UDP-based attacks with forged source addresses persist all the way through today. Aged computer systems with known vulnerabilities continued to be connected to the Internet and are readily transformed into attack bots.

The picture of attacks is also becoming more ominous. Previously attributed to 'hackers' it was quickly realised that a significant component of these hostile attacks had criminal motivations. The progression from criminal actors to state-based actors is also entirely predictable, and we are seeing an escalation of this cyber warfare arena with the investment in various forms of exploitation of vulnerabilities being seen as part of a set of desirable national capabilities.

It appears that a major problem here is that collectively we are unwilling to make any substantial investment in effective defence or deterrence. The systems that we use on the Internet are overly trusting to the point of

irrational credulity. For example, the public key certification system used to secure web-based transactions is repeatedly demonstrated to be untrustworthy, yet that's all we trust. Personal data is continually breached and leaked, yet all we seem to want to do is increase the number and complexity of regulations rather than actually use better tools that would effectively protect users.

The larger picture of hostile attack is not getting any better. Indeed, it's getting very much worse. If any enterprise has a business need to maintain a service that is always available for use, then any form of in-house provisioning is just not enough to be able to withstand attack. These days only a handful of platforms are able to offer resilient services, and even then it's unclear whether they could withstand the most extreme of attacks. There is a constant background level of scanning and probing going on in the network, and any form of visible vulnerability is ruthlessly exploited. One could describe today's Internet as a toxic wasteland, punctuated with the occasional heavily defended citadel. Those who can afford to locate their services within these citadels enjoy some level of respite from this constant profile of hostile attack, while all others are forced to try and conceal themselves from the worst of this toxic environment, while at the same time aware that they will be completely overwhelmed by any large scale attack.

It's a sobering through that about one half of the world's population are now part of this digital environment. A more sobering thought is that many of today's control systems, such as power generation and distribution, water distribution, and road traffic control systems are exposed to the Internet. Perhaps even more of a worry is the increasing use of the Internet in automated systems that include various life support functions. The consequences of massive failure of these systems in the face of a sustained and damaging attack cannot be easily imagined.

The Internet of Billions of Tragically Stupid Things

What makes this scenario even more depressing is the portent of the so-called Internet of Things.

In those circles where Internet prognostications abound and policy makers flock to hear grand visions of the future, we often hear about the boundless future represented by this "Internet of Things. This phrase encompasses some decades of the computing industry's transition from computers as esoteric pieces of engineering affordable only by nations, to mainframes, desktops, laptops, handhelds, and now wrist computers. Where next? In the vision of the Internet of Things we are going to expand the Internet beyond people and press on with using billions of these chattering devices in every aspect of our world.

What do we know about the "things" that are already connected to the Internet?

Some of them are not very good. In fact some of them are just plain stupid. And this stupidity is toxic, in that their sometime inadequate models of operation and security affects others in potentially malicious ways. Doubtless if such devices were constantly inspected and managed we might see evidence of aberrant behaviour and correct it. But these are unmanaged devices that are all but invisible. There are the controller for a web camera, the so-called "smart" thin in a smart television, or what controls anything from a washing machine to a goods locomotive. Nobody is looking after these devices.

When we think of an Internet of Things we think of a world of weather stations, web cams, "smart" cars, personal fitness monitors and similar. But what we tend to forget is that all of these devices are built upon layers of other people's software that is assembled into a product at the cheapest possible price point. It may be disconcerting to realise that the web camera you just installed has a security model that can be summarised with the phrase: "no security at all", and its actually offering a view of your house to the entire Internet. It may be slightly more disconcerting to realise that your electronic wallet is on a device that is using a massive compilation of open source software of largely unknown origin, with a security model that is not completely understood, but appears to be susceptible to be coerced into being a "yes, take all you want".

It would be nice to think that we've stopped making mistakes in code, and from now on our software in our things will be perfect. But that's hopelessly idealistic. It's just not going to happen. Software will not be perfect.

It will continue to have vulnerabilities. It would be nice to think that this Internet of Things is shaping up as a market where quality matters, and consumers will select a more expensive product even though its functional behaviour is identical to a cheaper product that has not been robustly tested for basic security flaws. But that too is hopelessly naive.

The Internet of Things will continue to be a market place where the compromises between price and quality will continue to push us on to the side of cheap rather than secure. What's going to stop us from further polluting our environment with a huge and diverse collection of programmed unmanaged devices with inbuilt vulnerabilities that will be all too readily exploited? What can we do to make this world of these stupid cheap toxic things less stupid and less toxic? Workable answers to this question have not been found so far.

The Next Ten Years

The silicon industry is not going to shut down anytime soon. It will continue to produce chips with more gates, finer tracks and more stacked layers for some years to come. Our computers will become more capable in terms of the rage and complexity of the tasks that they will be able to undertake.

At the same time, we can expect more from our network. Higher capacity certainly, but also greater levels of customisation of the network to our individual needs.

However, I find it extremely challenging to be optimistic about security and trust in the Internet. We have made little progress in this areas over the last ten years and there is little reason to think that the picture will change in the next ten years. If we can't fix it, then, sad as it sounds, perhaps we simply need to come to terms with an Internet jammed full of tragically stupid things.

However, beyond these broad-brush scenarios, it's hard to predict where the Internet will head. Technology does not follow a pre-determined path. It's driven by the vagaries of an enthusiastic consumer market place that is readily distracted by colourful bright shiny new objects, and easily bored by what we quickly regard as commonplace.

What can we expect from the Internet in the next ten years that can outdo a pocket-sized computer that can converse with me in a natural language? That can offer more than immersive 3D video in outstanding quality? That can bring the entire corpus of humanity's written work into a searchable database that can answer any of our questions in mere fractions of a second?

Personally, I have no clue what to expect from the Internet. But whatever does manage to capture our collective attention I am pretty confident that it will be colourful, bright, shiny, and entirely unexpected!

Disclaimer

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

Author

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

www.potaroo.net