The Internet Protocol Journal

June 2023 Volume 26, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.protocoljournal.org

ISSN 1944-1134

From The Editor

Twenty-five years ago, we published the first issue of *The Internet Protocol Journal* (IPJ). Since then, 87 issues for a total of 3,316 pages have been produced. Today, IPJ has about 20,000 subscribers all around the world. In the early days of IPJ, most of our readers preferred the paper edition, but over time preferences have shifted steadily to a situation where only some 1,200 print subscribers remain. The rest are downloading the PDF version. This shift in reading habits is likely related to the changes in technology that have taken place in the last 25 years. Lower costs and higher-resolution displays and printers, as well as improvements in Internet access technologies, have made the online "experience" a lot better than it was in 1998.

In this issue, we will first look at two areas of work taking place in the *Internet Engineering Task Force* (IETF). The *Application-Layer Traffic Optimization* (ALTO) protocol aims to make network state such as topology, link availability, routing policies, and path cost metrics information available to applications in a standardized manner. The next article concerns the thorny topic of *tracking* of users and their devices on the Internet. This area is complex, with many potential solutions, including the use of randomized *Media Access Control* (MAC) addresses as described by members of the *MAC Address Device Identification for Network and Application Services* (MADINAS) Working Group in the IETF.

Our final article is a look back at the last 25 years of Internet technology development. As we did with our 10th anniversary issue in 2008, we asked Geoff Huston to provide an overview of the many changes that have taken place in this period. At the end of his article, you will find a list of previously published articles from IPJ on numerous aspects of Internet technologies. All back issues are, of course, available from our website.

Let me take this opportunity to thank all the people who make IPJ possible. We are grateful to all our sponsors and donors, without whose generous support this publication would not exist. Our authors deserve a round of applause for carefully explaining both established and emerging technologies. They are assisted by an equally insightful set of reviewers and advisors who provide feedback and suggestions on every aspect of our publications process. The process itself relies heavily on two individuals: Bonnie Hupton, our copy editor, and Diane Andrada, our designer. Thanks go also to our printers and mailing and shipping providers. Last, but not least, our readers provide encouragement, suggestions, and feedback. This journal would not be what it is without them.

—Ole J. Jacobsen, Editor and Publisher
ole@protocoljournal.org

The IETF ALTO Protocol

Optimizing Application Performance by Increasing Network Awareness

by Qin Wu, Mohamed Boucadair, and Jordi Ros-Giralt

In today's Internet, network-related information (for example, topology, link availability, routing policies, and path cost metrics) are usually hidden from the application layer. As a result, endpoints make network-unaware decisions that may lead to suboptimal service placement and selection decisions, sometimes resulting in poor user experience and unnecessary inter-*Internet Service Provider* (ISP) traffic. Previous approaches to this problem space have considered snooping on the lower layers to determine the state and capabilities of the network, but such techniques require applications to be aware of lower-layer components (for example, routing protocols) and, furthermore, if left unspecified, can potentially overload key network resources.

To overcome this challenge, it is necessary to gather and expose network state information (for example, the bandwidth and latency properties between two network endpoints) to applications that do not interact directly with their underlying network protocols, without increasing the risk of network service disruption. For instance, empowered with such information, service providers can safely optimize the placement of their applications in locations of the network that provide higher capacity and lower latency to the clients they intend to serve. Similarly, with such information, client applications can also optimize the selection of the server instances they decide to attach to, while relying upon a variety of cost metrics.

This article provides an overview of how the *Internet Engineering Task Force* (IETF) *Application-Layer Traffic Optimization* (ALTO) protocol enables applications with improved network awareness to overcome these challenges, and reports on some of the implementations and deployments of the ALTO protocol.

The ALTO Approach and Architecture

The IETF ALTO protocol defines a client/server network service that applications can use to gain insightful information about the current state of the network. As defined in the base protocol[1], each ALTO server maintains a "my-Internet" view of the network it represents. In its simplest form, this view consists of a set of endpoints and costs between pairs of endpoints for each possible cost type (for example, hop count, latency, or bandwidth). An application seeking to gain this information to make optimized decisions can use an ALTO client to connect to an ALTO server using the *Hypertext Transfer Protocol* (HTTP)-based protocol defined in the ALTO base specification.

The ALTO protocol uses a *Representational State Transfer* (RESTful) design and encodes its requests and responses using *JavaScript Object Notation* (JSON) objects.

An ALTO request carries a set of source-destination endpoints and a cost type. The triggered ALTO response provides the cost value for each given source-destination endpoint. To improve scalability (for example, to reduce the load of an ALTO server) and privacy (for example, to avoid revealing sensitive topology information), ALTO introduces the concept of *groups*, which specify sets of endpoints that are close to each other from a network connectivity standpoint. In larger-scale neworks, this aggregation leads to greater scalability without losing critical information. A group may be represented as an IP prefix, a *Point of Presence* (PoP), a type of access connectivity (wireless, fiber, etc.), an *Autonomous System* (AS), or a set of ASes. The entity that operates an ALTO server, called the *ALTO Service Provider*, is responsible for assigning a unique *Provider-defined Identifier* (PID) to each group.

Another generalization of the endpoint object is enabled using the concept of *Abstract Network Element* (ANE). This concept provides an abstract representation of a component in a network that handles data packets and whose properties can potentially affect the end-to-end performance of an application[14]. ANEs can include not only endpoints, but also switches and routers that connect them.

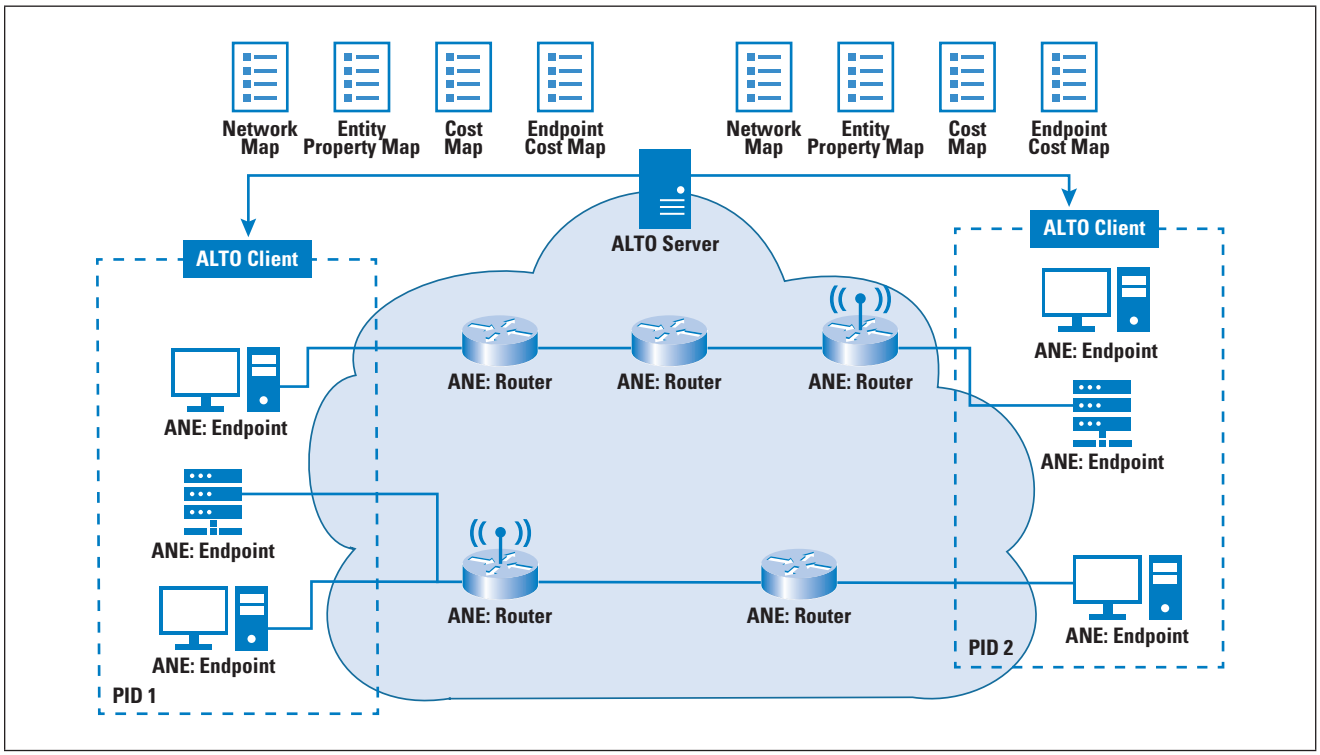
Figure 1, on the next page, depicts the main ALTO abstract objects involved in a network. In this figure, application endpoints are represented as ANEs clustered in two groups with provider-defined identifiers "PID1" and "PID2." An endpoint can select to communicate with another endpoint based on the network properties that the ALTO server exposes. For instance, in a *Content Delivery Network* (CDN), ANEs correspond to client and server hosts, and a specific content (for example, a movie) is in general replicated in more than one server instance. A client host can decide to retrieve the content by selecting the server instance that provides the higher communication bandwidth according to the exposed ALTO information. Each of the abstract objects that are illustrated in Figure 1 is further elaborated in the following sections.

ALTO Maps

An ALTO server organizes the network information using the concept of *maps*. Maps can be constructed from physical information, logical information, or a combination thereof. ALTO supports four types of maps, as shown in Figure 1:

- The *Network Map* lists all the endpoint groups that the ALTO server tracks. This map includes PIDs that uniquely identify each group.
- The *Entity Property Map* describes the properties of each ANE in the network, including the geolocation or the connectivity type (for example, fiber or wireless) of an ANE.
- The *Cost Map* provides the cost information (for example, hop count, latency, or bandwidth) between each pair of PIDs enclosed in the network map, where a PID identifies a group of endpoints.
- The *Endpoint Cost Map* provides finer-grained cost information between specific endpoints.

Figure 1: Base ALTO Abstract Objects



ALTO Extensions

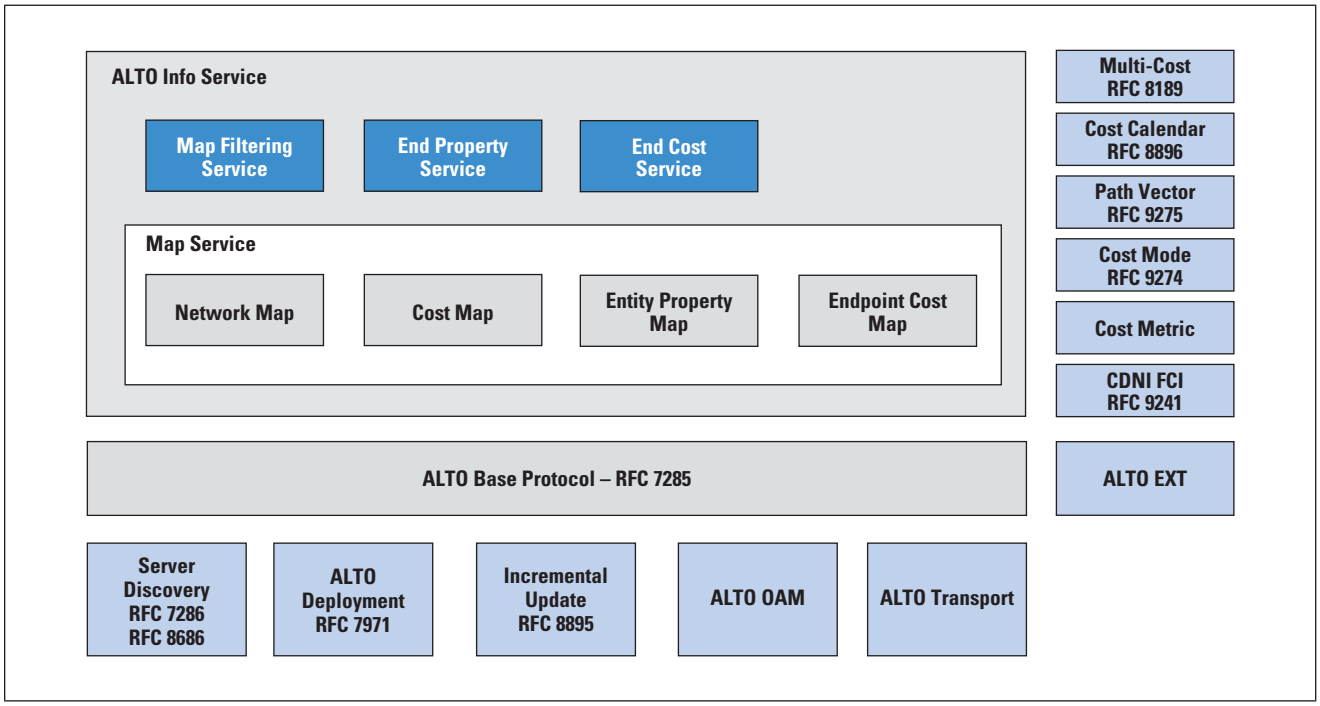
In addition to its base abstractions and maps, the ALTO protocol supports the following extensions to enable a richer network-aware application experience:

- The *Information Resource Directory* (IRD) lists the services an ALTO server provides and the locations from where you can access these services.
- The *Cost Calendar* provides a set of cost values as a function of time, allowing applications to know not only where to connect to, but also when.
- Incremental Updates using *Server-Sent Events* (SSEs) allow an ALTO server to expose cost values as delta updates, reducing the amount of server-client data exchanged.
- The CDNI Advertisement exposes a CDNI *Footprint and Capability Advertisement Interface* (FCI)[26].
- The *Path Vector extension* exposes the set of ANEs along the path between two endpoints and the performance properties of these ANEs.
- *The Extended Performance Cost Metrics* enrich ALTO with advanced metrics such as network one-way delay, one-way delay variation, one-way packet-loss rate, hop count, and bandwidth.
- The *Entity Properties* generalize the concept of ALTO endpoint properties by presenting them as entity property maps.

Figure 2 shows the ALTO protocol core services as they are documented by the IETF as well as some of the related ALTO documents. The previously mentioned ALTO extensions are marked with a light-blue shading. The core services are organized as part of the *ALTO Information Service consisting of the Map Filtering, End Property,* and *End Cost* services, along with the *Map Service*, which is itself broken into separate map services as previously described. All of the services are dependent on the base protocol, which is documented in [1]. The ALTO protocol is enhanced through *Server Discovery*[5, 25], and extensions for *Incremental Updates*[9], *Operations and Management* (OAM)[23], and support for carrying the ALTO protocol over more modern transport protocols[22]. The practical understanding of how you can use the ALTO protocol together with a set of deployment recommendations is documented in [13].

Additional ALTO features, for example, cost manipulation[7, 8], are shown on the right side of Figure 2.

Figure 2: Overview of ALTO Core and Extensions



History of ALTO

The ALTO Working Group was established in 2008 with an initial charter to develop a request/response protocol that would allow hosts to extract enough state information from a network to make optimized server selection decisions. The working group's first charter focused on the optimization of *Peer-to-Peer* (P2P) applications, with the first four RFCs introducing the problem statement[11] and requirements[4], the base protocol[1], and support for server discovery[5].

The working group was then rechartered in 2014 to support a broader set of applications that included CDNs and data centers.

That stage led to the development of five RFCs: Deployment recommendations[13], protocol extensions for reducing the volume of on-the-wire data exchange[7, 9], server discovery for multi-domain environments[25], and a cost calendar capability to allow applications to identify the optimal times to connect to a service[8].

The current ALTO Working Group charter was approved in 2021 with the goal to focus on three operational areas: (1) support for modern transport protocols such as HTTP/2 and HTTP/3[22]; (2) development of OAM mechanisms[23], and (3) collection of deployment experiences[24]. These three areas constitute the current highest priorities of the ALTO Working Group.

Four additional RFCs that had originated from the second charter have also been published since then: (1) support of property maps for generalized entities[10], (2) a new *Footprint and Capabilities Advertisement Interface* (FCI) protocol for CDNI[12], (3) a new *Internet Assigned Numbers Authority* (IANA) registry for tracking cost modes supported by ALTO[3], and (4) extensions to the cost map and ALTO property map services to allow the application to identify optimized paths[14].

ALTO Deployments and Implementations

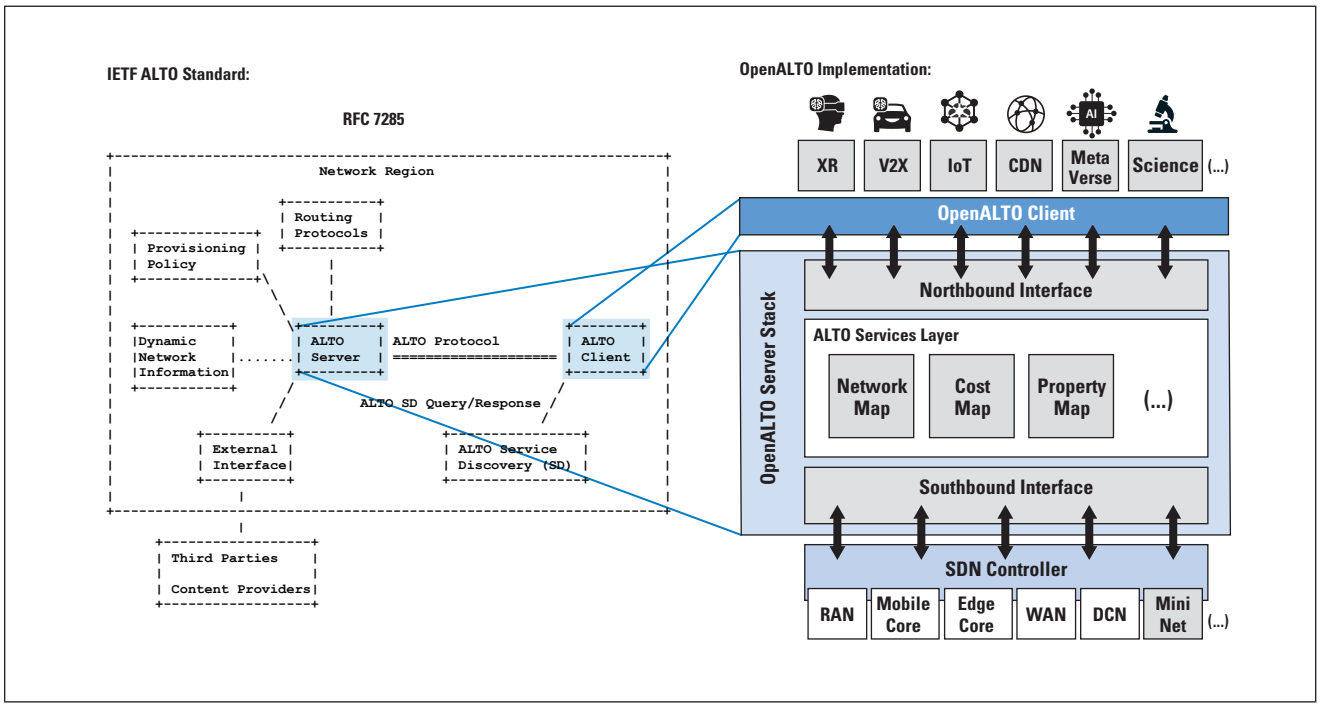
The ALTO base protocol was first implemented by Korea Telecom[16], NEC[17], Benocs[18], Alcatel-Lucent Bell Labs[19], and Nokia. An open-source implementation of the ALTO stack was also made available via the *OpenDaylight* (ODL) Project[6]. Starting in 2020, China Mobile, Tencent[20], and Telefonica[21] have been actively involved in ALTO and initiated trials in their mobile networks and CDNs. Qualcomm Technologies, Inc. also joined the ALTO effort in 2021 with a focus on evaluating the fit of ALTO for exposing network state information in the context of edge computing. At the time of this writing (2023), two new deployments of ALTO are being initiated to support the networks from CERN (LHCONE) in Europe and the Network Research Platform in the United States[27].

A further open-source initiative, the *OpenALTO* Project[2], was initiated in 2021 to provide a standalone implementation of the ALTO stack independently of the ODL Project. The OpenALTO Project is an initiative spawning from the IETF ALTO Working Group, which focuses on developing an open-source implementation of the ALTO specifications, including the latest Internet Drafts that have been moved to Working Group Last Call to support modern transport protocols[22] and OAM[23]. As shown in Figure 3, the architecture maps the IETF ALTO server and ALTO client onto the OpenALTO software stack as follows:

- The OpenALTO Server stack includes three core building blocks: The Application-facing Interface, the Network-facing Interface, and the ALTO Services Layer.
- The Application-facing Interface provides an *Application Programming Interface* (API) that applications can query to retrieve the state of the network.

- The Network-facing Interface implements a variety of network plugins to support the retrieval of network state information; each plugin supports a different type of network. To facilitate the development of OpenALTO, this interface also includes plugins for simulation and emulation environments such as *Mininet*.
- The ALTO Services Layer provides the core ALTO functions by implementing [1]. This layer currently includes the Network Map, the Cost Map, and the Property Map services.
- The OpenALTO client is a thin layer that implements the HTTP-based client-side protocol described in [1] and [9]. The ALTO client is installed as a library in the same device in which the application is being run, and the application uses it to retrieve the network state from the ALTO server.

Figure 3: Mapping of RFC 7285 Entities onto the OpenALTO Software Architecture



Future Perspectives

The ALTO protocol initially started with the goal of supporting the optimization of P2P applications in 2008, then evolved to incorporate extensions for the support of CDNs in 2014, and today it is well-positioned to support the requirements of new advanced edge computing applications such as augmented reality, vehicle networks, and the metaverse, among others. Because this new class of applications requires stringent *Quality of Experience* (QoE) performance, the ALTO protocol becomes a key component to enable collaborative application/network schemes.

Specifically, ALTO contributes to the optimization of service placement and selection decisions based on the communication properties of the network. In this regard, and as its current charter is being finalized, proposals are being made to extend the protocol towards supporting edge computing applications in three possible areas: (1) extending ALTO metrics to include information about the compute resources (for example, *Central Processing Unit* [CPU], *Graphics Processing Unit* [GPU], memory, and storage) found in the distributed edge computing network, (2) incorporating protocol semantics for the sharing of state between ALTO servers in multi-domain networking environments, helping applications gain a global end-to-end view of the network, and (3) potentially incorporating information about the level of trust offered by each ANE along a communication path to improve the security of new advanced applications such as the metaverse.

Beyond the IETF, several other *Standards Development Organizations* (SDOs) such as the *3rd Generation Partnership Project* (3GPP) are also investigating solutions for exposing network capabilities to enable the optimization of new advanced applications. These solutions can naturally take advantage of ALTO, and there is the potential for IETF technology to become an important enabler of Internet capabilities demanded by developments arising in other SDOs. To enable these cross-SDO synergies, the ALTO protocol needs to be further socialized inside and outside the IETF with a focus on illustrating how it can provide the intended exposure features.

Future directions of the ALTO protocol are currently being discussed in the WG mailing list (**`https://mailarchive.ietf.org/arch/browse/alto/`**). The WG welcomes your participation to help identify the key priorities towards supporting the newly arising edge computing applications.

References and Further Reading

[1] Richard Alimi, Ed., Reinaldo Penno, Ed., Richard Yang, Ed., Sebastian Kiesel, Stefano Previdi, Wendy Roome, Stanislav Shalunov, and Richard Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol," RFC 7285, September 2014.

[2] OpenALTO Project, available at **`https://github.com/openalto`**

[3] Mohamed Boucadair and Qin Wu, "A Cost Mode Registry for the Application-Layer Traffic Optimization (ALTO) Protocol," RFC 9274, July 2022.

[4] Sebastian Kiesel, Stefano Previdi, Martin Stiemerling, Richard Woundy, and Richard Yang, "Application-Layer Traffic Optimization (ALTO) Requirements," RFC 6708, September 2012.

[5] Sebastian Kiesel, Martin Stiemerling, Nico Schwan, Michael Scharf, and Haibin Song, "Application-Layer Traffic Optimization (ALTO) Server Discovery," RFC 7286, November 2014

[6] ODL ALTO, available at: **`https://wiki.opendaylight.org/display/ODL/ALTO`**

[7] Sabine Randriamasy, Wendy Roome, and Nico Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)," RFC 8189, October 2017.

[8] Sabine Randriamasy, Richard Yang, Qin Wu, Lingli Deng, and Nico Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar," RFC 8896, November 2020.

[9] Wendy Roome and Richard Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)," RFC 8895, November 2020.

[10] Wendy Roome, Sabine Randriamasy, Richard Yang, Jingxuan Zhang, and Kai Gao, "An Extension for Application-Layer Traffic Optimization (ALTO): Entity Property Maps," RFC 9240, July 2022.

[11] Jan Seedorf and Eric Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement," RFC 5693, October 2009.

[12] Jan Seedorf, Richard Yang, Kevin Ma, Jon Peterson, and Jingxuan Zhang, "Content Delivery Network Interconnection (CDNI) Footprint and Capabilities Advertisement Using Application-Layer Traffic Optimization (ALTO)," RFC 9241, July 2022.

[13] Martin Stiemerling, Sebastian Kiesel, Michael Scharf, Hans Seidel, and Stefano Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations," RFC 7971, October 2016.

[14] Qin Wu, Richard Yang, Young Lee, Dhruv Dhody, Sabine Randriamasy, and Luis Contreras, "ALTO Performance Cost Metrics," Internet Draft, Work in Progress, **draft-ietf-alto-performance-metrics-28**, March 2022.

[15] Richard Barnes, "Use Cases and Requirements for JSON Object Signing and Encryption (JOSE)," RFC 7165, April 2014.

[16] Choongul Park, Yeongil Seo, Kun-youl Park, and Youngseok Lee, "The concept and realization of context-based content delivery of NGSON," in *IEEE Communications Magazine*, Volume 50, No. 1, pp. 74–81, January 2012.

[17] Marcus Schöller, Martin Stiemerling, Andreas Ripke, and Roland Bless, "Resilient deployment of virtual network functions," 2013 *5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops* (ICUMT), Almaty, Kazakhstan, 2013.

[18] Enric Pujol, Ingmar Poese, Johannes Zerwas, Georgios Smaragdakis, and Anja Feldmann, "Steering Hyper-Giants' Traffic at Scale," In *Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies* (CoNEXT '19). Association for Computing Machinery, New York, 2019.

[19] Michael Scharf, Thomas Voith, Wendy Roome, Bob Gaglianello, Moritz Steiner, Volker Hilt, and Vijay K. Gurbani, "Monitoring and abstraction for networked clouds," *16th International Conference on Intelligence in Next Generation Networks,* Berlin, Germany, 2012.

[20] Yuhang Jia, Yunfei Zhang, Richard Yang, Gang Li, Yixue Lei, Yunbo Han, and Sabine Randriamasy, "MoWIE for Network Aware Application," Internet Draft, Work in Progress, **draft-huang-alto-mowie-for-network-aware-app-05**, November 2022.

[21] Jingxuan Zhang et al., "Sextant: Enabling Automated Network-aware Application Optimization in Carrier Networks," 2021 *IFIP/IEEE International Symposium on Integrated Network Management* (IM), Bordeaux, France, 2021.

[22] Roland Schott, Richard Yang, Kai Gao, Lauren Delwiche, and Lachlan Keller, "The ALTO Transport Information Publication Service," Internet Draft, Work in Progress, **draft-ietf-alto-new-transport-07**, April 2023.

[23] Jingxuan Zhang, Dhruv Dhody, Kai Gao, Roland Schott, and Q. Ma, "YANG Data Models for the Application-Layer Traffic Optimization (ALTO) Protocol," Internet Draft, Work in Progress, **draft-ietf-alto-oam-yang-06**, April 2023.

[24] ALTO IETF Wiki on Collecting Deployment Experiences, available at: **https://wiki.ietf.org/en/group/ALTO/deployment**

[25] Sebastian Kiesel and Martin Stiemerling, "Application-Layer Traffic Optimization (ALTO) Cross-Domain Server Discovery," RFC 8686, February 2020.

[26] Jan Seedorf, Jon Peterson, Stafano Previdi, Ray van Brandenburg, and Kevin Ma, "Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics," RFC 8008, December 2016.

[27] Jacob Dunefsky, Mahdi Soleimani, Ryan Yang, Jordi Ros-Giralt, Mario Lassnig, Inder Monga, Frank K. Würthwein, Jingxuan Zhang, Kai Gao, and Y. Richard Yang, "Transport control networking: optimizing efficiency and control of data transport for data-intensive networks," *ACM SIGCOMM*, August 2022.

QIN WU is an expert on Network Management Architecture with Huawei's Data Communication in China. He is also responsible for enterprise networking innovation and standards work such as IoT, Security, SD-WAN, and Edge Computing. Involved in strategic standards development, engaging with some related open-source projects such as FD.io and ONAP for more than 10 years, Qin has held various positions in IETF, ITU-T, and CCSA. He has over 16 years of experience on network architecture and protocol design, starting from mobility management, performance measurement, IPTV to SDN, NFV, network management automation and YANG, telemetry, AIOPs, etc. Currently he focuses on promoting digital twin networking and Network and Application collaboration. He used to chair the IETF L3SM and L2SM working groups in the OPS area; he currently chairs the ALTO Working Group in the Transport area, serves as a member of the OPS-DIR Directorate, and has coauthored more than 52 RFCs spanning six IETF areas (OPS, SEC, RTG, TSV, RAI, and INT). He received his PhD degree of Control Theory and Engineering from Nanjing University of Science and Technologies. Qin is a member of the Internet Architecture Board.
E-mail: **bill.wu@huawei.com**

MOHAMED BOUCADAIR is a Senior Network Architect within the "Network of the Future" team in Orange Innovation. He worked at the Orange corporate division, where he was responsible for making recommendations on the evolution of IP/MPLS core networks. Mohamed is the author/editor of several books, including *Design Innovation and Network Architecture for the Future Internet* (ISBN: 9781799876465), *Emerging Automation Techniques for the Future Internet* (ISBN: 9781522571469), *Redesigning the Future of Internet Architectures* (ISBN: 9781466683716), *Solutions for Sustaining Scalability in Internet Growth* (ISBN: 978-1466643055), *IP Telephony Interconnection Reference: Challenges, Models and Engineering* (ISBN: 978-1439851784), *Recent Advances in Providing QoS and Reliability in Future Internet Backbone* (ISBN: 978-1617618581), *Inter-Asterisk Exchange (IAX) Deployment Scenarios in SIP-Enabled Networks* (ISBN: 978-0470770726), and *Service Automation and Dynamic Provisioning Techniques in IP/MPLS Environments* (ISBN: 978-0470018291).
E-mail: **mohamed.boucadair@orange.com**

JORDI ROS-GIRALT is a Director of Engineering at Qualcomm Europe, Inc., where he leads the high-performance networking team as part of AI Research focusing on the area of accelerating application performance for 5G, 6G, and the Edge Cloud. Jordi has published upwards of 75 articles in scientific conferences and journals, and is the inventor and developer of several communication network algorithms and technologies, most of which have been included in commercial products. Jordi received his PhD in Computer Science, an MBA from the University of California, and a BSc in Telecommunications Engineering from the Barcelona Tech University (UPC).
E-mail: **jros@qti.qualcomm.com**

The Impact of Randomized Layer-2 Addresses on Privacy and Applications

by Carlos J. Bernardos, Juan-Carlos Zuñiga, Jerome Henry, and, Alain Mourad

Wi-Fi technology has revolutionized communication and become the preferred technology and sometimes the only networking technology used by devices such as smartphones, tablets, and *Internet-of Thing* (IoT) devices.

On the other hand, Internet privacy is becoming a huge concern, as more and more mobile devices are connecting to the Internet. This ubiquitous connectivity, together with not-very-secure protocol stacks and the lack of proper education about privacy, make it very easy to track/monitor the location of users and/or eavesdrop on their physical and online activities. The cause of this situation has many factors, such as the vast digital footprint that users leave on the Internet; for example: information sharing on social networks, the cookies that browsers and servers use to provide a better navigation experience, connectivity logs that allow tracking of a user's Layer 2 *Media Access Contro*l (L2 MAC) or Layer 3 (L3) address, web trackers, etc., and/or the weak (or even null in some cases) authentication and encryption mechanisms used to secure communications.

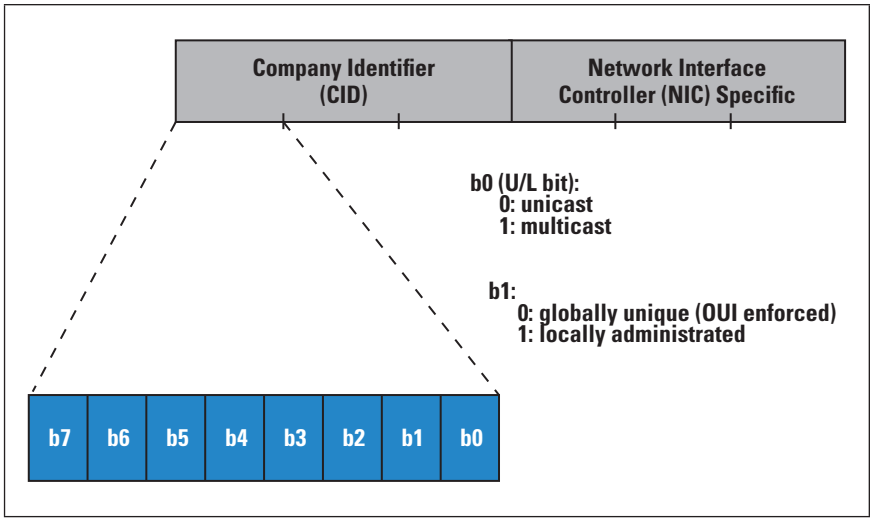
This privacy concern affects all layers of the protocol stack, from the lower layers involved in the actual access to the network (for example, you can use the L2 and L3 addresses to obtain a user's location) to higher-layer protocol identifiers and user applications[1]. In particular, IEEE 802 MAC addresses have historically been an easy target for tracking users[2]. Attackers who are equipped with surveillance equipment can "monitor" Wi-Fi packets and track the activity of Wi-Fi devices. After the association between a device and its user is made, identifying the device and its activity is sufficient to deduce information about what the user is doing, without the user's consent.

IEEE 802.11 (Wi-Fi) interfaces, as any other kind of IEEE 802-based network interface, like Ethernet (that is, IEEE 802.3), have a Layer 2 address, also referred to as the MAC address, which anybody who can receive the signal transmitted by the network interface can see. Figure 1 shows the format of these addresses.

A third party, such as a passive device listening to communications in the same network, can easily observe MAC addresses. In an 802.11 network, a station exposes its MAC address in two different situations:

- While unassociated and actively scanning for available networks, the MAC address is used in the *Probe Request* frames that the device sends (aka IEEE 802.11 STA).
- After it is associated to a given *Access Point* (AP), the MAC address is used in frame transmission and reception, as one of the addresses used in the address fields of an IEEE 802.11 frame.

Figure 1: IEEE 802 MAC Address Format



MAC addresses can be either universally or locally administered. A MAC address is identified as being locally administered when the second-last significant bit of the most significant octet of the address (the U/L bit) is set. The MAC address is identified as globally unique when the U/L bit is unset.

A universally administered address is uniquely assigned to a device by its manufacturer (and is called the *burned-in address*). Most physical devices are provided with a universally administered address, which is composed of two parts: (i) the *Company Identifier* (CID), which is the first three octets in transmission order, identified by the organization that issued it, and (ii) the *Network Interface Controller* (NIC)-specific address, which is the following three octets, assigned by the organization that manufactured the controller, in such a way that the resulting MAC address is globally unique. Since universally administered MAC addresses are by definition globally unique, when a device uses this MAC address to transmit data—especially over the air—it is relatively easy to track this device by simple medium observation. This possibility poses a privacy concern[3] when the device is directly associated to a single user (for example, smartphones, etc.).

Locally administered addresses can override the burned-in address, and they can be set up by either the network administrator or the *Operating System* (OS) of the device to which the address pertains. This feature allows you to generate local addresses without the need for any global coordination mechanism to ensure that the generated address is still unique within the local network. You can use this feature to generate random addresses, which decouple the globally unique identifier from the device and thereby make it more difficult to track a user device from its MAC/L2 address[4]. There are initiatives at the IEEE 802 and other organizations to specify ways in which these locally administered addresses should be assigned, depending on the use case.

To reduce the risks of correlation between a device activity and its owner, multiple vendors have started to implement *Randomized and Changing MAC* (RCM) addresses. With this scheme, an end device implements a different RCM over time when exchanging traffic over a wireless network. By randomizing the MAC address, the persistent association between a given traffic flow and a single device is made more difficult, assuming no other visible unique identifiers are in use.

However, such address changes may affect the user experience and the efficiency of legitimate network operations. For a long time, network designers and implementers relied on the assumption that a given machine in a network implementing IEEE 802 technologies would be represented by a unique network MAC address that would not change over time, despite the existence of tools to flush out the MAC address to bypass some network policies. When this assumption is broken, elements of network communication may also break.

For example, sessions established between the end device and network services may be lost and packets in translation may suddenly be without a clear source or destination. If multiple clients implement fast-paced RCM rotations, network services may be over-solicited by a small number of stations that appear as many clients.

At the same time, some network services rely on the client station providing an identifier, which can be the MAC address or another value. If the client implements MAC rotation but continues sending the same static identifier, then the association between a stable identifier and the station continues despite the RCM scheme. There may be environments where such continued association is desirable, but others where the user privacy has more value than any continuity of network service state.

Application and Network Scenarios That RCM Can Affect

Device identity is important in scenarios where the network needs to know the device or user identity in order to offer, operate, and maintain certain valid services. Currently, many use cases and applications make an implicit assumption that a device is represented by an IEEE 802 L2 permanent and unique MAC address. This assumption is being used in both control- and data-plane functions and protocols. RCM breaks this assumption. This paradigm shift requires updating applications to function across MAC address changes.

When a device changes its MAC address, other devices on the same LAN may fail to recognize that the same machine is attempting to communicate with them. Additionally, multiple layers implemented at upper layers have been designed with the assumption that each node on the LAN, using these services, would have a MAC address that would stay the same over time (a *persistent* MAC address).

This assumption sometimes adds to the *Personally Identifiable Information* (PII) confusion, for example in the case of *Authentication, Association, and Accounting* (AAA) services authenticating the user of a machine and associating the authenticated user to the device MAC address. Other services focus solely on the machine, for example, the *Dynamic Host Configuration Protocol* (DHCP), but still expect each machine to use a persistent MAC address, for example to re-assign the same IP address to a returning device. Changing the MAC address may disrupt these services and the user experience.

The impact of using a persistent or a randomized and changing MAC address very much depends on the environment where the device operates (that is, the use case), on the presence and nature of other devices in the environment, and on the type of network the device is communicating through. Therefore, a device can use a MAC address that can persist over time if trust with the environment is established, or that can be temporal if that address is going to be used as an identity for a service in an environment where trust has not been established. Note that this trust is not binary, and it ranges from: (i) full trust: environments where a personal device establishes a trust relationship and can share a persistent device identity with the access network devices, without the fear of that identity being shared beyond the L2 broadcast domain; (ii) selective trust: environments where the device may not be willing to share a persistent identity with some elements of the Layer 2 broadcast domain but may be willing to do it with other elements; and (iii) zero trust: environments where the device may not be willing to share any persistent identity with any local entity reachable through the AP and may express a temporal identity to each of them.

This trust relationship naturally depends on the relationship between the user of the personal device and the operator of the service. Thus, it is useful to enumerate some scenarios (which can be easily translated into use cases) where the use of RCM might have an impact:

- *Residential settings under the control of the user:* this case is typical of a home network with Wi-Fi in the LAN and an Internet connection. In this environment, traffic over the Internet does not expose the MAC address if it is not copied to another field before routing happens. The user controls the wire segment within the broadcast domain, and this segment, therefore, is usually not at risk of hosting an eavesdropper. Full trust is typically established at this level among users and with the network elements. The device trusts the AP and all Layer 2 domain entities beyond the AP. However, unless the user has full access control over the physical space where the Wi-Fi transmissions can be detected, there is no guarantee that an eavesdropper would not be observing the communications. As such, it is common to assume that, even in this environment, full trust cannot be achieved.

- *Managed residential settings:* examples of this type of environment include shared living facilities and other collective environments where an operator manages the network for the residents. The over-the-air exposure is similar to that of a home. A number of devices larger than in a standard home may be present, and the operator may be requested to provide IT support to the residents. Therefore, the operator may need to identify the activity of a device in real time, but may also need to analyze logs so as to understand a past reported issue. For both activities, a device identification associated with the session is needed. Full trust is often established in this environment, at the scale of a series of a few sessions, not because it is assumed that no eavesdropper would observe the network activity, but because it is a common condition for the managed operations.
- *Public guest networks:* public hotspots, such as in shopping malls, hotels, stores, train stations, and airports, are typical of this environment. The guest network operator may be legally mandated to identify devices or users or may have the option to leave all devices and users untracked. In this environment, trust is commonly not established with any element of the Layer 2 broadcast domain (zero trust model by default).
- *Enterprises (with Bring Your Own Device [BYOD]):* users may be provided with corporate devices or may bring their own. The devices are not directly under the control of a corporate IT team. Trust may be established as the device joins the network. Some enterprise models mandate full trust; others, considering the BYOD nature of the device, allow selective trust.
- *Managed enterprises:* in this environment, users are typically provided with corporate devices, and all connected devices are managed, for example through a *Mobile Device Management* (MDM) profile installed on the device. Full trust is created as the MDM profile is installed.

Ongoing Efforts/Approaches Regarding RCM

Practical experiences of RCM in live devices helped researchers fine-tune their understanding of attacks against randomization mechanisms[5]. At IEEE 802.11 these research experiences eventually formed the basis for a specified mechanism introduced in the IEEE 802.11aq in 2018, which recommends mechanisms to avoid pitfalls when using randomized MAC addresses[6].

More recent developments include turning on MAC randomization in mobile operating systems by default, which affects the ability of network operators to personalize or customize services[7]. Therefore, follow-on work in the IEEE 802.11 mapped effects of a potentially large uptake of randomized MAC identifiers on many commonly offered operator services in 2019[8].

In the summer of 2020, this work resulted in two new standards projects with the purpose of developing mechanisms that do not decrease user privacy and enable an optimal user experience when the MAC address of a device in an *Extended Service Set* is randomized or changes[9] and user privacy solutions applicable to IEEE Std 802.11[10].

The IEEE 802.1 Working Group has also published a specification that defines a local MAC address space structure, known as the *Structured Local Address Plan* (SLAP). This structure designates a range of local MAC addresses for protocols using a CID assigned by the IEEE Registration Authority. Another range of local MAC addresses is designated for assignment by administrators. The specification recommends a range of local MAC addresses for use by IEEE 802 protocols[11].

Work within the IEEE 802.1 Security Task Group on privacy recommendations for all IEEE 802 network technologies has also looked into general recommendations on identifiers, reaching the conclusion that temporary and transient identifiers are preferable in network technology designs if there are no compelling reasons of service quality for a newly introduced identifier to be permanent. This work has been specified in the recently published IEEE P802E: "Recommended Practice for Privacy Considerations for IEEE 802 Technologies"[12]. The IEEE P802E specification will form part of the basis for the review of user privacy solutions applicable to IEEE Std 802.11 (aka Wi-Fi) devices as part of the RCM[7] efforts.

Currently, two task groups in IEEE 802.11 are addressing issues related to RCM:

- The IEEE 802.11bh Task Group, looking at mitigating the repercussions that RCM creates on 802.11 networks and related services, and
- The IEEE 802.11bi Task Group, which will define modifications to the IEEE Std 802.11 MAC specification to specify new mechanisms that address and improve user privacy.

At the *Wireless Broadband Alliance* (WBA), the *Testing and Interoperability Working Group* has been looking at the issues related to MAC address randomization and has identified a list of potential impacts of these changes to existing systems and solutions, mainly related to Wi-Fi identification. As part of this work, WBA has documented a set of use cases that a Wi-Fi Identification Standard should address in order to scale and achieve longer-term sustainability of deployed services. A first version of this document has been liaised with the IETF as part of the *MAC Address Device Identification for Network and Application Services* (MADINAS) activities through the "Wi-Fi Identification in a post MAC Randomization Era v1.0" paper[13].

Several IP address assignment mechanisms such as the IPv6 *Stateless Address Auto-Configuration* (SLAAC) techniques[14] generate the *Interface Identifier* (IID) of the address from its MAC address (via EUI64), which then becomes visible to all IPv6 communication peers. This feature potentially allows for global tracking of a device at L3 from any point on the Internet. Besides, the prefix part of the address provides meaningful insights into the physical location of the device in general, which together with the MAC address-based IID, makes it easier to perform global device tracking.

Some solutions might mitigate this privacy threat, such as the use of temporary addresses[15] and opaque IIDs[16,17]. Additionally, [18] proposes an extension to DHCPv6 that allows a scalable approach to link-layer address assignments where preassigned link-layer address assignments (such as by a manufacturer) are not possible or unnecessary. [19] proposes extensions to DHCPv6 protocols to enable a DHCPv6 client or relay to indicate a preferred SLAP quadrant to the server, so that the server may allocate MAC addresses in the quadrant requested by the client or relay.

Not only can you use MAC and IP addresses for tracking purposes, but some DHCP options allow you to also carry unique identifiers. These identifiers can enable device tracking even if the device administrator takes care of randomizing other potential identifications such as link-layer addresses or IPv6 addresses. [20] introduces anonymity profiles, designed for clients that wish to remain anonymous to the visited network. The profiles provide guidelines on the composition of DHCP or DHCPv6 messages, designed to minimize disclosure of identifying information.

Existing Solutions

One possible solution is to use 802.1X with *Wi-Fi Protected Access 2/3* (WPA2/WPA3). At the time of association to a Wi-Fi access point, 802.1X authentication coupled with WPA2 or WPA3 encryption schemes allows for the mutual identification of the client device or of the user of the device and an authentication authority. The authentication exchange is protected from eavesdropping. In this scenario, you can obfuscate the identity of the user or the device from external observers. However, the authentication authority is in most cases under the control of the same entity as the network access provider, thus making the identity of the user or device visible to the network owner. This scheme is therefore well-adapted to enterprise environments, where a level of trust is established between the user and the enterprise network operator.

A different approach is the Wireless Broadband Alliance *OpenRoaming* standard, which introduces an intermediate trusted relay between local venues and sources of identity. The federation structure also extends the type of authorities that can be used as identity sources (compared to the traditional enterprise-based 802.1X scheme for Wi-Fi), and facilitates the establishment of trust between a local venue and an identity provider.

Such a procedure dramatically increases the likelihood that one or more identity profiles for the user or the device will be accepted by a local venue. At the same time, authentication does not occur to the local venue, thus offering the possibility for the user or device to keep their identity obfuscated from the local network operator, unless that operator specifically expresses the requirement to disclose such identity (in which case the user has the option to accept or decline the connection and associated identity exposure). The OpenRoaming scheme therefore seems well-adapted to public Wi-Fi and hospitality environments, allowing for the obfuscation of the identity from unauthorized entities, while also permitting mutual authentication between the device or the user and a trusted identity provider.

It is also worth mentioning that most evolved client device OSes already offer RCM schemes, enabled by default (or easy to enable) on client devices. With these schemes, the device can change its MAC address, when not associated, after using a given MAC address for a semi-random duration window. These schemes also allow for the device to manifest a different MAC address in different *Service Set Identifiers* (SSIDs). Different OSes follow slightly different approaches, which are also evolving with the new releases. Such a randomization scheme enables the device to limit the duration of exposure of a single MAC address to observers. In the IEEE 802.11-2020 specification, MAC address rotation is not allowed during a given association session, and thus rotation of MAC address can occur only through disconnection and reconnection.

Ongoing Work in the IETF

The MADINAS Working Group in the IETF is addressing documentation of the services that may be affected by RCM and evaluation of possible solutions to maintain the quality of user experience and network efficiency in the presence of RCM, while user privacy is reinforced.

The group will generate documents regarding the state of affairs of RCM, and a *Best Current Practices* (BCP) document recommending a means to reduce the impact of RCM on the documented use cases while ensuring that the privacy achieved with RCM is not compromised. For scenarios where device identity stability is desirable, the BCP document will recommend existing protocols that you can use to protect the request and exchange of identifiers between the client and the service provider.

The MADINAS Working Group is focused on coordination with other IETF Working Groups (for example, DHC and IntArea). In addition, it actively liaises with other relevant organizations, such as IEEE 802 and the Wireless Broadband Alliance. The objective is to coordinate on the different recommendations, as well as planning potential follow-up activities within or outside the IETF.

It is expected that the outcome from the coordinated efforts among these standards organizations will enable the use of RCM in the different scenarios previously analyzed, providing both privacy and the operational characteristics about the quality of user experience and network efficiency that each one of the scenarios requires.

References

[1] Carlos J. Bernardos, Juan Carlos Zúñiga, and Piers O'Hanlon, "Wi-Fi Internet Connectivity and Privacy: Hiding your tracks on the wireless Internet," *IEEE Conference on Standards for Communications and Networking* (CSCN), October 2015.

[2] James Vincent, "London's bins are tracking your smartphone," *The Independent*, August 2013, **https://www.independent.co.uk/life-style/gadgets-and-tech/news/updated-london-s-bins-are-tracking-your-smartphone-8754924.html**

[3] Piers O'Hanlon, Joss Wright, and Ian Brown, "Privacy at the link layer," Contribution at W3C/IAB workshop on *Strengthening the Internet Against Pervasive Monitoring* (STRINT), February 2014. **https://www.w3.org/2014/strint/papers/35.pdf**

[4] Marco Gruteser and Dirk Grunwald, "Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis," *Mobile Networks and Applications*, Volume 10, No. 3, pp. 315-325, 2005.

[5] Jeremy Martin, Travis Mayberry, Colin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Eric C. Rye, and Dane Brown, "A Study of MAC Address Randomization in Mobile Devices and When It Fails," arXiv:1703.02874v2 [cs.CR], 2017.

[6] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802.11aq-2018 - IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Preassociation Discovery," IEEE 802.11, 2018.

[7] Institute of Electrical and Electronics Engineers (IEEE),"IEEE 802.11 Randomized and Changing MAC Addresses Study Group CSD on User Experience Mechanisms," doc.:IEEE 802.11-20/1346r1, 2020.

[8] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802.11 Randomized and Changing MAC Addresses Topic Interest Group Report," doc.:IEEE 802.11-19/1442r9, 2019.

[9] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802.11 Randomized and Changing MAC Addresses Study Group PAR on User Experience Mechanisms," doc.:IEEE 802.11-20/742r5, 2020.

[10] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802.11 Randomized and Changing MAC Addresses Study Group PAR on Privacy Mechanisms," doc.:IEEE 802.11-19/854r7, 2020.

[11] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802c-2017 - IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture—Amendment 2: Local Medium Access Control (MAC) Address Usage," IEEE 802c, 2017.

[12] Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802E-2020 - IEEE Recommended Practice for Privacy Considerations for IEEE 802 Technologies," IEEE 802E, 2020.

[13] Wide Band Alliance, "Wi-Fi Identification Scope for Liaising - In a post MAC Randomization Era," doc.:WBA Wi-Fi ID Intro: Post MAC Randomization Era v1.0 - IETF liaison, March 2020.

[14] Susan Thomson, Thomas Narten, and Tatuya Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862, September 2007.

[15] Richard Draves and Dave Thaler, "Default Router Preferences and More-Specific Routes," RFC 4191, November 2005.

[16] Fernando Gont, "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)," RFC 7217, April 2014.

[17] Fernando Gont, Alissa Cooper, Dave Thaler, and Will Liu, "Deprecating EUI-64 Based IPv6 Addresses," Internet Draft, Work in Progress, **draft-gont-6man-deprecate-eui64-based-addresses-00**, October 2013.

[18] Bernie Volz, Tomek Mrugalski, and Carlos J. Bernardos, "Link-Layer Address Assignment Mechanism for DHCPv6," RFC 8947, December 2020.

[19] Carlos J. Bernardos and Alain Mourad, "Structured Local Address Plan (SLAP) Quadrant Selection Option for DHCPv6," RFC 8948, December 2020.

[20] Christian Huitema, Tomek Mrugalski, and Suresh Krishnan, "Anonymity Profiles for DHCP Clients," RFC 7844, May 2016.

CARLOS J. BERNARDOS received a Telecommunication Engineering degree in 2003, and a PhD in Telematics in 2006, both from the University Carlos III of Madrid, where he works as a Full Professor. He teaches different undergraduate and masters degree courses, including the Master and Specialist in NFV and SDN. His research interests include IP mobility management, network virtualization, cloud computing, vehicular communications, and experimental evaluation of mobile wireless networks. He has published over 100 scientific papers in international journals and conferences. Carlos has been an active contributor to IETF since 2005, being co-author of more than 30 contributions and several standards, he has co-chaired the IETF P2PSIP and IPWAVE Working Groups, and he currently co-chairs the MADINAS Working Group and the Internet Area Directorate (INTDIR). E-mail: **cjbc@it.uc3m.es**

JUAN CARLOS ZÚÑIGA is a technology and business development trilingual leader with more than 20 years of international experience. He currently leads wireless standardisation and IP efforts within the Global Technical Standards team at Cisco Systems. He has extensive experience with heterogeneous Radio Access Networks and 4G/5G Core network cloud technologies. Juan is an active thought leader and contributor in different standards and industrial fora, such as IETF, IEEE 802, ISOC, ETSI, 3GPP SA/CT/RAN, ITU, W3C, and GS1. He has strong experience developing, managing, analyzing, and developing international intellectual property (IPR) patent portfolios. He has been granted patents for more than 70 USPTO/EPO inventions and over 100 published applications. Juan holds several recognitions and awards. E-mail: **juzuniga@cisco.com**

JEROME HENRY is a Principal Engineer in the Enterprise Infrastructure and Solutions Group at Cisco Systems. Jerome joined Cisco in 2012. Before that time, he consulted and taught about heterogeneous networks and wireless integration with the European Airespace team, which Cisco later acquired to become its main wireless solution. Jerome holds more than 150 patents, is a member of the IEEE, where he was elevated to Senior Member in 2013, and also represents Cisco in multiple Wi-Fi Alliance working groups. With more than 10,000 hours in the classroom, Jerome was awarded the IT Training Award Best Instructor silver medal. He is based in Research Triangle Park, North Carolina. E-mail: **jerhenry@cisco.com**

ALAIN MOURAD is an award-winning innovator with over 20 years of experience in the Research and Development of wireless technologies spanning four generations of cellular systems (3G/4G/5G and heading towards 6G). He currently heads the Future Wireless Europe Research and Innovation Lab at InterDigital in London (UK). Prior to InterDigital, Alain was a Principal Engineer at Samsung Electronics Research and Development and a Senior Engineer at Mitsubishi Electric R&D Centre Europe, where he was active in the specification of wireless standards (3GPP, IEEE802, DVB, and ATSC). E-mail: **Alain.Mourad@interdigital.com**

Twenty-Five Years Later

by Geoff Huston, APNIC

The Internet is not quite as young and spritely as you might have thought. Apple's iPhone, released in 2007, is now 16 years old, and YouTube is an ageing teenager at 18 after its initial release in 2005, and these two examples are relatively recent additions to the Internet. The first web browser, Mosaic, was released some 30 years ago in 1993. Going back further, the Internet emerged from its early *Advanced Research Projects Agency* (ARPA) roots in the form of the *National Science Foundation Network* (NSFNET) in 1986. At the start of 1983, the *ARPA Network* (ARPANET) had a flag day and switched over to use the *Transmission Control Protocol* (TCP). Going back further, in 1974 Vint Cerf and Bob Kahn published the first academic paper describing the protocol and the underlying architectural framework of a packet-switched network that became the Internet. This achievement was built upon earlier foundations, where numerous efforts in the late 1960s showed the viability of a *packet-switched* approach to computer networking. These packet-switched networking efforts included a program led by Donald Davies at the UK National Physics Laboratory, an effort in the US in the form of an ARPA project led by Larry Roberts, and Louis Pouzin's work in France with the CYCLADES network. This work, in turn, has some of its antecedents in work by Paul Baran at the RAND Corporation on distributed communications and packet-switched networks, published between 1960 and 1964. The Internet has managed to accumulate a relatively lengthy pedigree.

And it has been a wild ride. The Internet has undergone numerous cycles of economic boom and bust, each of which is right up there with history's finest episodes of exuberant irrational mania. It has managed to trigger a comprehensive restructuring of the entire global communications enterprise and generated a set of changes that have already altered the way in which we now work and play. That's quite a set of achievements in just 25 years!

We should start this exploration of our past some 25 years ago in 1998 at the time of publication of the first edition of the *Internet Protocol Journal*. At that time, any lingering doubts about the ultimate success of the Internet as a global communications medium had been thoroughly dispelled. The Internet was no longer just a research experiment, or an intermediate way stop on the road to adoption of the *Open Systems Interconnection* (OSI) framework. By 1998 there was nothing else left standing in the data communications landscape that could serve our emerging needs for data communications. The *Internet Protocol* (IP) was now the communications technology for the day, if not for the coming century, and the industry message at the time was a clear one that said: "adopt the Internet into every product and service or imperil your entire future in this business."

No longer did the traditional telecommunications enterprises view the Internet with some polite amusement or even overt derision. It was now time for a desperate scramble to be part of this revolution in one of the world's major activity sectors. The largest enterprises in this sector, the old-world ex-monopoly telcos, had been caught wrong-footed in one of the biggest changes of the industry for many decades, and this time the concurrent wave of deregulation and competition meant that the future of the communications industry was being handed over to a small clique of Internet players.

By the early 2000s, the Internet had finally made it into the big time. The job was apparently done, and the Internet had prevailed. But then came a new revolution, this time in mobility services, where after numerous clumsy initial efforts by others, the iPhone entered the market with a seamless blend of sleek design and astounding capability. The mobile carriage sector struggled to keep up with the new levels of rapacious demand for Internet-based mobile data. The Internet then took on the television networks, replacing the incumbent broadcast and cable systems with streaming video. But the story is not over by any means. Communications continues to drive our world, and the Internet continues to evolve and change.

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 technical evolution of the Internet appears to be no exception, and this story contains these same forms of unanticipated turns and twists.

Rather than offer a set of unordered observations about the various changes and developments over the past 25 years, I will use the traditional protocol stack model as a template, starting with the underlying transmission media, then looking at IP, the transport layer, then applications and services, and closing with a look at the business of the Internet.

Transmission

It seems like a totally alien concept these days, but the *Internet Service Provider* (ISP) business of 1998 was still based around the technology of dial-up modems. The state-of-the-art of modem speed had been continually refined, from 9600 bps to 14.4 kbps, to 28 kbps, to finally 56 kbps, squeezing every last bit out of the phase amplitude space contained in an analogue voice circuit. Analogue modems were capricious, constantly being superseded by the next technical refinement, unreliable, difficult for customers to use, and on top of that, they were slow! Almost everything else on the Internet had to be tailored to download reasonably quickly over a modem connection. Web pages were carefully composed with compressed images to ensure a rapid download, and plain text was the dominant medium as a consequence. It could only get better.

The evolution of access networks was initially one that exposed the inner digital core of the network out to the edges. The first approach was *Integrated Services Digital Network* (ISDN), where the underlying digitised voice circuit was drawn out to the network edge. At 64 kbps, this level of improvement was inadequate, and the next major step was to use *Digital Subscriber Line* (DSL) technology. DSL used the last mile of the network for an analogue channel, but instead of running a single low-speed bearer signal, DSL layered a large collection of individual bearer signals into the single circuit, performing a form of frequency division multiplexing on the basic analogue circuit in a trellis framework. DSL relied on the combination of the telephone company's efforts to operate the copper access circuits within a base level of signal quality and noise suppression, and the modem industry's continual incremental improvements in digital-signal-processing capability. Surprisingly, DSL achieved speeds of tens of megabits per second through these legacy copper access networks. However, DSL was largely an interim holding position while the search for a viable business model that could underwrite the costs of deployment and use of an open fibre-based access networks was underway.

The transition into fixed-wire access networks based on fibre-optic cable continues. The challenge is not in finding a suitable technology for fibre, but one of finding a suitable business model than can sustain the necessary capital investment in replacing the existing copper-based infrastructure. Some national communities used a model of a public-sector program, such as the *National Broadband Network* program in Australia, while others have remained as dedicated private-sector activities, and others have taken a hybrid approach with some level of local public-sector incentives being added into a private-sector program. The issue here is that fixed wire residential access networks do not offer compelling investment opportunities in most cases, with the high initial capital costs and the generally inadequate levels of take-up across the dwellings passed by the access infrastructure acting as disinhibitory factors. It is often the case that a residential community cannot support multiple access network deployments, bringing up the related issue of local access monopolies and the challenge of permitting some level of competitive access across a single physical access network. Nevertheless, fibre access rollouts continue across many parts of the world, and the transition of the wired copper network into a fibre access network capable of sustaining hundreds of megabits per connection is still progressing, seemingly in spite of the financial barriers that exist in many scenarios.

Mobile

The mobile network has experienced a completely different evolution, and for many years now the mobile sector has been demand-driven. The first mobile data service networks, introduced in the 1980s, were little more than low-speed digital encoders working across a single voiceband circuit. These 1G networks typically delivered a 2.4-kbps data download capacity.

The next generation of mobile services, 2G, was used in the 1990s. It was still predominately a voice service, and while it could theoretically support data access at speeds of 100 kbps, this data-transfer rate was largely aspirational, and the mobile network was predominantly used by the *Short Message Service* (SMS) as an adjunct to voice. The intersection of the Internet and mobile services occurred with the introduction of 3G mobile services. The 3G architecture could push IP connectivity directly out to the handset, supporting data-transfer speeds of 1–2 Mbps. This network capability, coupled with a new generation of handsets, first with the BlackBerry in 2002 and then the iPhone in 2007, transformed the mobile service into a mass-market consumer service. The high margins available from the service captured the attention of the traditional voice industry, and the result was a large-scale opening up of radio spectrum to create an Internet access environment that quickly rivalled the wire-line access market in size, but totally outpaced it in terms of revenue margins. This massive escalation of demand created further pressures on the capacity of the mobile system, and in 2009 the mobile sector introduced 4G services, opening up more spectrum bands, and also adding *Multiple-Input Multiple Output* (MIMO) to the mobile device to achieve greater deliverable capacity to each connected device. Over time these services were to deliver peak download speeds of 50 to 100 Mbps. The industry was also selling hundreds of millions of mobile devices per year. 4G dispensed with circuit-switched services, and it exclusively used packet switching. In 2018 5G was introduced. 5G can be supported over more spectrum bands, including a high-band millimetre spectrum at 24–47Ghz. These higher carrier frequencies permit multi-gigabit data services, but they come at a cost of higher density of base-station towers to compensate for the lower propagation distances.

Wi-Fi and Bluetooth

A second radio technology that has also transformed the Internet emerged in 1998, and it could be argued that it has become so fundamental that it has weaved itself so naturally into our environment that it all but disappeared. The combination of low-power radio systems and unlicensed radio spectrum allocation, or *Wi-Fi*, and subsequently *Bluetooth*, has been transformational. The combination of efficient battery technology, computer chips that operate with low power consumption, and the unwiring of the last few meters in the home and office completely changed our collective of technology, and it is only because of our desire to use products that are portable, unobtrusively wearable, and powerful enough to be useful that the component technologies such as batteries and processors have been pushed in this direction over this period. While large bands of radio spectrum space have been allocated to cellular mobile service operators, the intensity of use and the utility of use of radio spectrum peaks in the unlicensed spectrum space used by Wi-Fi and Bluetooth. It could be argued that the economic value of these unlicensed spectrum bands exceeds the exclusively licensed cellular radio systems by orders of magnitude. It could also be argued that the untethering of the last meter of the Internet transformed the Internet, and digital technologies in general, from a specialist pursuit into the consumer product space.

In the 1990s we described the effort to simplify the use of technology through the term "plug and play." Wi-Fi was the critical technical development that made that term irrelevant by removing any need for the plug, or the socket for that matter!

Satellite

Mobile data services, Wi-Fi, and Bluetooth really revolutionised the Internet, taking it from a "destination you visit" to an "always-on utility in your pocket." The Internet was now a set of applications on a device that went with you everywhere. Always available, always connected, no matter where you might be or what you might be doing. But that was not exactly the full truth. Head out into remote country far enough, or head onto the world's oceans, and your connection options quickly disappeared, leaving only somewhat expensive satellite-based services.

These satellite services have been in operation since the early 1960s, but the high launch costs, limited capacity, and competing interests of terrestrial providers have meant that these services were often operated at the margins of viability. The best example is Motorola's *Iridium* project of the late 1990s, where even before the entire service constellation of satellites was launched, the $5B Iridium project was declared bankrupt. *Starlink*, a recent entrant in the satellite service area, is using a constellation of some 4,000 low-earth-orbiting spacecraft and appears so far to have been able to break through this financial barrier. Using reusable launch vehicles, smaller (and lighter) satellites, transponder arrays on board, and a new generation of digital-signal-processing capabilities, Starlink is in a position to offer retail access services of 100 Mbps or more to individual customers. The low altitude of the spacecraft means that the Starlink service competes directly with terrestrial access services in terms of performance. The introduction of inter-spacecraft laser links means that the system can provide a service in any location, and the limiting factor, as with the Iridium effort decades ago, is obtaining the necessary clearances and licenses to have customers located in the respective national geographies. Starlink is certainly revolutionary in terms of capacity, coverage, and cost. The questions are whether it is sufficiently revolutionary and whether it can scale up to provide a high-capacity service to hundreds of millions of users. At this point in time these questions are not easy to answer, but the limitations inherent in *Low Earth Orbit* (LEO)-based services point to a potential advantage in terrestrial-based access networks. Nevertheless, Starlink is redefining the Internet access market space in many countries, and setting price/performance benchmarks that their terrestrial competitors now have to match.

If we move away from access networks to look at the changes in the "core" of the Internet over the past 25 years, then once more we can see a dramatic change. In 1998, the Internet was constructed using the margins of oversupply in the telephone networks. In 1998, the core infrastructure of most ISPs was still being built by leasing telephone trunk supergroups (E-1 and T-1 circuits, and then E-3 and T-3 as capacity needs escalated, and then OC-1 circuits).

While it was not going to stop here, squeezing even more capacity from the network was now proving to be a challenge; 622-Mbps IP circuits were being deployed, although many of them were constructed using 155-Mbps *Asynchronous Transfer Mode* (ATM) circuits using router-based load balancing to share the IP load over four of these circuits in parallel. Gigabit circuits were just around the corner, and the initial exercises of running IP over 2.5-Gbps *Synchronous Digital Hierarchy* (SDH) circuits were being undertaken in 1998.

In some ways 1998 was a pivotal year for IP transmission. Until this time, IP was still just one more data application that was positioned as just another customer of the telco's switched-circuit infrastructure. This telco infrastructure was designed and constructed primarily to support telephony. From the analogue voice circuits to the 64K digital circuit through to the higher-speed trunk bearers, IP had been running on top of the voice network infrastructure. Communications infrastructure connected population centres where there was call volume. The Internet had different demands. Internet traffic patterns did not mirror voice traffic, and IP performance is sensitive to every additional millisecond of delay. Constraining the Internet to the role of an overlay placed on top of a voice network was showing signs of stress, and by 1998 things were changing. The Internet had started to make ever larger demands on transmission capacity, and the driver for further growth in the network infrastructure was now not voice, but data. It made little sense to provision an ever-larger voice-based switching infrastructure just to repackage it as IP infrastructure, and by 1998 the industry was starting to consider just what an all-IP high-speed network would look like, building an IP network all the way from the photon in a fibre-optic cable all the way through to the design of the Internet application.

Fibre Optics

At the same time, the fibre-optic systems were changing with the introduction of *Wave Division Multiplexing* (WDM). Older fibre equipment with electro-optical repeaters and *Plesiochronous Digital Hierarchy* (PDH) multiplexors allowed a single fibre pair to carry around 560 Mbps of data. WDM allowed a fibre pair to carry multiple channels of data using different wavelengths, with each channel supporting a data rate of up to 10 Gbps. Channel capacity in a fibre strand was between 40 and 160 channels using *Dense WDM* (DWDM). Combined with the use of all-optical amplifiers, the most remarkable part of this entire evolution in fibre systems is that a cable system capable of an aggregate capacity of a terabit can be constructed today for much the same cost as a 560-Mbps cable system of the mid-1990s. That's a cost-efficiency improvement of a factor of one million in a decade. The drive to deploy these high-capacity DWDM fibre systems was never based on expansion of telephony. The explosive growth of the industry was all about supporting the demand for IP. So, it came as no surprise that at the same time as the demand for IP transmission was increasing there was a shift in the transmission model where instead of plugging routers into telco switching gear and using virtual point-to-point circuits for IP, we started to plug routers into wavelengths of the DWDM equipment and operate all-IP networks in the core of the Internet.

DWDM is not the only technology that has fundamentally changed these core transmission systems in the past 25 years. Two further technologies have been transformational in the fibre-optic area. The first is the use of optical amplifiers. *Erbium Doped Fibre Amplifiers* provide a highly efficient means of signal amplification without the need to convert the signal back unto a digital form and then passing it back through a digital/analogue converter to modulate the next-stage laser driver. This technology has allowed fibre systems to support terabit-per-second capacity without necessarily having to integrate terabit-per-second digital systems. The second fundamental change was a switch in signal modulation from a basic on/off signal into a signal modulation technique that uses signal-phase amplitude and polarity to increase the total capacity of a wavelength within a fibre strand. *Digital Signal Processors* (DSPs) offered the key technology here, and as we improve on the track width of conductor tracks in these processors we can increase the gate count on a single chip, thereby allowing support for more complex signal manipulation algorithms. These algorithms can be used to increase the sensitivity of the DSP function. In 2010, we were using 40-nm track silicon chips in DSPs, supporting *Polarization Mode Quadrature Phase Shift Keying* (PM-QPSK), which allowed a cable to operate with 100-Gbps data rates in a single wavelength, or an aggregation of 8 Tbps in a fibre strand. In 2023, DSPs now use 5-nm tracks, which can support PCS-144QAM modulation of a base 190-Gbaud signal, which can support 2.2-Tbps data rates per wavelength, or 105-Tbps total capacity per fibre strand. A 12-strand cable would have a total capacity of 1.2 Pbs.

Such very-high-performance fibre cable systems are generally used in submarine cable systems to link data centres between continents. In data-centre contexts and other terrestrial scenarios, we are now using 200- and 400-G per wavelength fibre system as the common technology base. The major outcome is that, in general, transmission is no longer a scarce resource. It is in every sense of the term an abundant commodity. There is no sense in rationing access to communications capacity, be it short or long haul. This change is a major one not only in the economic framework of the communications industry, but also in phrasing the way in which we use communications. In a scarce system, we tend to use "just-in-time" delivery mechanisms, passing content across the communications system only when it is needed, while an abundant system allows us to use "just-in-case" delivery mechanisms, causing a dramatic impact on the architecture of the Internet. Indeed, this extraordinary increase in the underlying capacity of our communications infrastructure through the past 25 years is perhaps the most significant change in the entire landscape of the Internet, as we will see when we look at content networking.

Network Management

In network operations, 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) more than a quarter of a century ago, and despite its security weaknesses, its inefficiency, its incredibly irritating use of *Abstract Syntax Notation One* (ASN.1), and its application in sustaining some forms of *Distributed Denial-of-Service* (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 *Network Configuration Protocol* (NETCONF) and *Yet Another Next Generation* (YANG) data modelling languages are attempting to pull this area of configuration management into something a little more usable than *Command-Line Interface* (CLI) scripts driving interfaces on switches.

At the same time, we are seeing orchestration tools such as *Ansible*, *Chef*, *Network Automation and Programmability Abstraction Layer with Multivendor* (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. The desired endpoint of a fully automated network-management framework is still far from our reach. Surely it must be feasible to feed an adaptive autonomous control system with the network 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 service-level objectives of the network? Where is the driverless car for driving networks? Maybe the next 10 years will get us there.

The Internet Layer

If our transmission systems have been subject to dramatic changes in the past quarter century, then what has happened at the IP layer over the same period?

First, we need to consider the "elephant" in the Internet layer room. One fundamental change at the Internet level of the protocol stack was meant to have all happened some 20 years ago, and that's the transition to IP version 6. Twenty-five years ago, in 1998, we were forecasting that we would have consumed all the remaining unallocated IPv4 addresses by around 2025. That estimate gave us slightly more than 25 years, so there was no particular sense of urgency. We didn't need to ring the emergency bell or raise any alarms. The overall aim was to proceed in an orderly manner. Things took a different course because we failed to appreciate the true impact of the shift of the Internet to mobile devices. All of a sudden, we were dealing with an Internet with billions of users, using billions of new mobile devices, and our comfortable predictions of a stately and steady rundown of the IPv4 address pools were discarded about as quickly as you could say the word "iPhone." From "all the time in the world" we reached "no time left to do anything" within a year.

In the 5-year period between 2005 and 2010, when mobile services exploded in volume, the total count of allocated IP addresses rose from 1.5B addresses to 3.1B, from a total address pool of 3.7B addresses. The network had doubled in size, and the time left to complete the transition had shrunk from more than 20 years to a little over 1!

At that point, all the plans for an orderly transition were discarded, and many network administrators scrambled to obtain IPv4 addresses, further depleting the IPv4 pools. The central pool of IPv4 addresses, operated by the *Internet Assigned Numbers Authority* (IANA), was exhausted in February 2011. The *Asia Pacific Network Information Centre* (APNIC) depleted its IPv4 pool in April of that year, the *Réseaux IP Européens Network Coordination Centre* (RIPE NCC) 18 months later, the *Latin America and Caribbean Network Information Centre* (LACNIC) in 2014, and the *American Registry for Internet Numbers* (ARIN) in 2015. We had expected that this situation would motivate network operators to hasten their plans for IPv6 deployment, yet, perversely, that did not happen. Less than 1% of the Internet user base was using IPv6 in 2011. Five years later, as each of the *Regional Internet Registries* (RIRs) ran down their remaining pools of IPv4 addresses, this Internet-wide IPv6 user count had increased to just 5%. In 2023, the process is still underway, and some 35% of the Internet user base has IPv6 capability. I'm not sure anyone is willing to predict how long this anomalous situation of running the IPv4 Internet "on an empty tank" will persist.

NATs

How has the Internet managed to continue to operate, and even grow, without a supply of new IPv4 addresses? In a word, the answer is "NATs." While the *Network Address Translator* (NAT) concept received little fanfare when it was first published, it has enjoyed massive deployment over the past 25 years, and today NATs are ubiquitous. The application architecture of the Internet has changed, and we are now operating a client/server framework. Servers have permanent IP addresses, while clients "borrow" a public IPv4 address to complete a transaction and return it back to a common pool when they are done. Time-sharing IP addresses, and also using the 16-bit source port field in TCP and the *User Datagram Protocol* (UDP), has managed to extend the IPv4 address space by some 20 bits, making the IPv4+NAT address space up to a million times larger than the original 32-bit IPv4 address space. In practice, the story is a little more complicated than that, and some very large service providers have reached logistical limits in using NATs to compensate for the exhaustion of IPv4 addresses. This situation has motivated these providers to transition to a dual-stack mode of operation, and they are relying on a dual-stack host behaviour that prefers to use IPv6 when possible, thus relieving the pressure on the IPv4 NAT functions

NATs have prompted a significant change at the IP level in changing the default assumption about the semantics of an IP address. An IP address is no longer synonymous with the persistent identity of the remote party, but it has assumed the role of an ephemeral session token.

The leisurely pace of the IPv6 transition is partly due to this altered role of addresses, as we no longer require every connected device to have a persistently assigned globally unique IP address.

IPv6 and NATs are not the only areas of activity in the Internet layer in the past 25 years. We have tried to change many parts of the Internet layer, but interestingly, few, if any, of the proposed changes have managed to gain any significant traction out there in the network. The functions performed at the Internet layer of the protocol stack are no different from those of 25 years ago. IP Mobility, Multicast, and *IP Security* (IPSec) are largely Internet layer technologies that have failed to gain significant levels of traction in the marketplace of the public Internet.

QoS

Quality of Service (QoS) was a "hot topic" in 1998, and it involved the search for a reasonable way for some packets to take some form of expedited path across the network, while other packets took an undifferentiated path. We experimented with various forms of signalling, packet classifiers, queue-management algorithms, and interpretations of the *Type of Service* bits in the IPv4 packet header, and we explored the QoS architectures of *Integrated and Differentiated Services* in great detail. However, QoS never managed to get established in mainstream Internet service environments. In this case, the Internet took a simpler direction, and in response to not enough network capacity we just augmented the network to meet demand.

Again, this is an aspect of the altered mindset when the communication system shifts from scarcity and rationing to one of abundance. We have given up installing additional intricate mechanisms in the network, in host protocol stacks, and even in applications to negotiate how to share insufficient network capacity. So far, the simple approach of just adding more capacity to the network has prevailed, and QoS remains largely unused.

MPLS

The switch from circuit switching to packet switching has never managed to achieve universal acceptance. We have experimented with putting circuits back into the IP datagram architecture in various ways, most notably with the *Multi-Protocol Label Switching* (MPLS) technology. This technology used the label-swapping approach that was previously used in X.25, *Frame Relay* and ATM virtual circuit-switching systems, and it created a collection of virtual paths from each network ingress to each network egress across the IP network. The original idea was that in the interior of the network you no longer needed to load up a complete routing table into each switching element, and instead of performing destination-address lookup you could perform a much smaller, and hopefully faster, label lookup. This performance differentiator did not eventuate and switching packets using the 32-bit destination address in a fully populated forwarding table continued to present much the same level of cost efficiency at the hardware level as virtual circuit label switching.

However, one aspect of MPLS and similar approaches has proved to be invaluable for many network operators. A general-purpose network utility has many disparate client networks, and a single packet-switched environment does not allow the network operator to control the way in which the common network resource is allocated to each client network. It also does not readily support segmentation of reachability. Virtual circuit overlays, such as MPLS, provide mechanisms to control resource allocation and constrain cross-network leakage, and for many network operators these reasons are adequate to head down an MPLS-like path for their network platform.

Routing

Moving sideways at this level of the protocol stack, we probably should look at the evolution of routing technologies. The early 1990s saw a flurry of activity in the routing space, and various routing protocols were quickly developed and deployed. By 1998 the conventional approach to routing was to use either *Intermediate System-to-Intermediate System* (IS-IS) or *Open Shortest Path First* (OSPF) as the interior routing protocol, and the *Border Gateway Protocol* (BGP) as the inter-domain routing protocol. This picture has remained constant right up to today. In some ways, it is reassuring to see a basic technology that can sustain a quite dramatic growth rate through many years of scaling, but in other ways it is less reassuring to see that the unresolved issues we had with the routing system in 1998 are largely still with us today.

The largest of these unresolved issues lies in the trust we place in the inter-domain routing system of the Internet. There is no overall orchestration of the routing system. Each network advertises reachability information to its adjacent networks and selects what it regards as the "best" reachability information from the set received from these same network peers. This mutual trust that each network places in all other networks can, and has, been abused in various ways. The effort to allow each routing entity to distinguish between what is a "correct" item of routing information and what is a "false" route has a rich history of initiatives that have faltered for one reason or another. The most recent effort in this space is built upon the foundations of the number system, and it uses the association of a public/private key pair with the current holders of addresses and autonomous system numbers, allowing these holders to issue signed authorities about the use of these number resources in the context of routing, and by coupling these authorities with the information being propagated in the routing system, the intention being that unauthorized use cases will be detected.

RPKI

This effort, the *Resource Public Key Infrastructure* (RPKI), has achieved some level of acceptance in the networking space, and in 2023 around one-third of all route objects have associated RPKI credentials. The work is still "in progress" because the more challenging aspect of this work is to associate verifiable credentials with the propagation route through a network that does not impose onerous burdens on the routing system and is not overly fragile in its operation.

The extended period where the routing system has operated in a state that essentially cannot be trusted has prompted the application layer to generate its own mechanisms of trust. These days it is largely left to *Transport Layer Security* (TLS) to determine whether a client has reached its intended server. Given that we have been unable to construct a secured routing system for many decades, the question arises whether there is still the same level of need for such a system that we had some 25 years ago, given that the application space sees this problem as largely solved through the close-to-ubiquitous use of TLS.

This tension between the Internet layer and the upper layers of the protocol stack is also evident in the way in which we have addressed the perennial issue of location and identity. One of the original simplifications in the IP architecture was to bundle the semantics of identity, location, and forwarding into an IP address. While that has proved phenomenally effective in terms of simplicity of applications and simplicity of IP networks, it has posed some serious challenges when considering mobility, routing, protocol transition, and network scaling. Each of these aspects of the Internet would benefit considerably if the Internet architecture allowed identity to be distinct from location. Numerous efforts have been directed at this problem over the past decade, particularly in IPv6, but so far, we really haven't arrived at an approach that feels truly comfortable in the context of IP. The problem we appear to have been stuck on for the past decade is that if we create a framework of applications that use identity as a rendezvous mechanism and use an IP layer that requires location, then how is the mapping between identity and location distributed in an efficient and suitably robust manner? The transport layer of the protocol stack has also looked at the same space and developed some interesting approaches, as we will see in the next section.

Transport

Back in 1998 the transport layer of the IP architecture consisted of UDP and TCP, and the network use pattern was around 95% TCP and 5% UDP. It has taken all of the intervening 25 years, but this picture has finally changed.

We have developed some new transport protocols in this period, such as the *Datagram Congestion Control Protocol* (DCCP) and the *Stream Control Transmission Protocol* (SCTP), which can be regarded as refinements of TCP to extend a flow-control mechanism to apply to datagram streams in the case of DCCP and a shared flow-control state over multiple reliable streams in the case of SCTP. However, in a world of transport-aware middleware that has been a constant factor over this period, the level of capability to actually deploy these new protocols in the public Internet is marginal at best. Firewalls do not recognize these more recent transport protocols, NATs and similar, and as a result, the prospects of wide-scale deployment of such protocols in the public Internet are not very good. We seem to be firmly stuck in a world of TCP and UDP.

TCP has proved to be remarkably resilient over the years, but as the network increases in capacity the ability of TCP to continue to deliver ever-faster data rates over distances that span the globe is becoming a significant issue. Much work has been done to revise the TCP flow-control algorithms so that they still share the network fairly with other concurrent TCP sessions yet can ramp up to multi-gigabit-per-second data-transfer rates and sustain those rates over extended periods of time. The mainstream TCP flow-control protocol has been shifting from the conventional Reno-styled protocol to CUBIC, which attempts to find a stable sending rate and then slowly add flow pressure to the network path to see if the network can support greater sending rates. The response to packet drop remains a dramatic rate drop, but not quite as dramatic as the rate halving of Reno, but nevertheless it is still a drop-sensitive ack-paced flow-control protocol.

However, the picture has changed with the introduction of the *Bottleneck Bandwidth and Round-Trip* (BBR) protocol. Driving the network into the point not only of network queue formation, but right to the point of queue overflow and packet loss, is a crude approach. The problem here is that packet loss represents a loss of feedback, and in a feedback-based flow-control protocol, this loss of feedback pushes the protocol into a space where it has to pull back its sending rate to re-establish a signal flow. BBR represents a different way of looking at flow control, and it attempts to drive the flow to the point of the onset of queue formation in the network rather than aiming at the point of queue collapse. This process reduces the latency of the flow and the cost of network switching equipment by reducing the very-high-speed fast memory buffer requirements.

This area is not the only one of new experimentation in changing the TCP congestion-control, paradigm. Another approach is being explored in the *Low Latency Low Loss Scalable* throughput initiative (L4S), which is looking at incorporating network signals into the flow-control algorithm. Here the packet switches use the *Explicit Congestion Notification* (ECN) signal in the IP header when standing queues start to form. The receiver of this signal is expected to back off its sending rate in a manner similar to packet loss. The advantage of this approach is that there is no loss of feedback signalling, and the flow reacts to the formation of congestion conditions rather than the end point of queue collapse. However, ECN requires the deployment of ECN-marking equipment, and the effort of synchronising network equipment and transport-protocol behaviours is far greater when compared to protocol-only approaches such as BBR.

Other initiatives in the transport space that are also worthy of note include Multipath TCP and QUIC.

The first of these initiatives is *Multipath TCP*. The observation here is based around the increasing ubiquity of both Wi-Fi and cellular radio services, and the configuration of most mobile devices to include the ability to access both of these networks.

In general, the choice of which network interface to use is a single decision made by the mobile platform for all active applications. When a usable Wi-Fi network is detected, the device will prefer to use that connection for all new connections because it is assumed that the Wi-Fi service will be cheaper for the user and will operate at a higher performance level. But if performance and resilience are issues, then can we allow a TCP session to use *all* the available networks at once, and optimise its use of these multiple network paths to the destination such that the total data throughput is optimised? This is the objective of Multipath TCP, where a single TCP session is broken into numerous sub-sessions, where each sub-session uses a different network path by using a different local network interface.

Multipath TCP allows separate TCP states to control the flows passing across each network path to optimise throughput. It also can permit flow migration, allowing a logical TCP flow to switch from one network path to another while preserving integrity. The interesting aspect of this behaviour is that the control of the multipath behaviours is, in the first instance, under the control of the application rather than the host platform. This response was an early one to recognize the increasing capacity and diversity in edge networks, and how we could respond to this situation at the transport session level.

QUIC

The second initiative, which for me is a fundamental change in transport capabilities and functions, is the introduction of the QUIC protocol. At its simplest level, you could say QUIC is a packaging of the combination of TCP and TLS into a UDP wrapping. However, I would suggest that such a description is well short of the mark. QUIC is in many ways a far more ambitious transport protocol, bringing transport to the point where it is better suited to the current application behaviour. QUIC is intended to improve the transport performance for encrypted traffic with faster session setup. QUIC allows for further evolution of transport mechanisms with support for *Remote Procedure Calls* (RPC). QUIC also has integral support for concurrent session multiplexing that avoids TCP head-of-line blocking. QUIC encrypts the payload data, but unlike TLS, QUIC also encrypts the control data (the equivalent of the TCP header) and explicitly avoids the emerging TCP ossification within the network by occluding the entirety of the control exchange from the network of the session. QUIC is address agile, in that it can react to network-level address renumbering in an active QUIC session, as can occur with the presence of NATs on the network path. You can implement QUIC in user space, so applications can control their own transport functions. There is no longer a dependence on the platform in terms of the quality of the implementation of the transport service. With QUIC the application exercises a comprehensive level of control of the way the application interacts with the network.

Numerous lessons can be drawn from the QUIC experience. Any useful public communications medium needs to safeguard the privacy and integrity of the communications that it carries.

The time when open protocols represented an acceptable compromise between efficiency, speed, and privacy are over, and these days all network transactions in the public Internet need to be protected by adequate encryption. The QUIC model of wrapping a set of transactions, including both data and control transactions between a client and a server, into an end-to-end encryption state represents a minimum level of functionality in today's networking environment.

Secondly, QUIC provides needed additional transport functionality. TCP and UDP represent just two points of transport functions within a broader spectrum of possible transport models. UDP is just too susceptible to abuse, so we have heaped everything onto TCP. The issue is that TCP was designed as an efficient single streaming protocol, and retrofitting multiple sessions, short transactions, remote procedure calls, reliable single-packet transactions, and shared congestion states have proved to be impossible to implement in TCP.

Applications are now dominant in the Internet ecosystem, while platforms and networks are being commoditised. We are seeing loss of patience with platforms that provide common transport services for the application that they host, and a new model where the application comes with its own transport service. Taking an even broader perspective, the context of the success of the Internet lies in shifting the responsibility for providing service from the network to the end system. This shifting allowed us to make more efficient use of the common network substrate and push the cost of this packetization of network transactions over to end systems.

It shifted the innovation role from the large and lumbering telco operators into the nimbler world of software. QUIC takes it one step further, and pushes the innovation role from platforms to applications, just at the time when platforms are declining in relative importance within the ecosystem. From such a perspective, the emergence of an application-centric transport model that provides faster services, a larger repertoire of transport models, and encompassing comprehensive encryption were inevitable developments.

We have pushed the responsibility for end-to-end authentication into the transport layer with the close-to-ubiquitous TLS. TLS layers themselves above TCP (or merges with the TCP-like function in the case of QUIC), and the client passes the name of the service it intends to connect with to the remote server. The server passes its public key to the client, and the client authenticates this key using its own trust anchors. The server and client then negotiate a session key and proceed with an encrypted session. TLS is robust in almost every respect. Its major weakness lies in the highly distributed trust model, where there are hundreds of different operators of trusted credentials (certification authorities) and thousands of various registration agents. These entities are placed in a highly trusted role, and they can never lie.

The problem is that they have proved to be corruptible occasionally. They typically operate using online services, and a successful attack against such platforms can be abused to allow the issuance of trusted public certificates. We have invested considerable time and effort in shoring up this trust framework, but at the same time we have been working to make these public key certificates a commodity rather than an expensive luxury. The introduction of free certification authorities has succeeded in making these certificates available to all, but at the same time the totally automated certificate issuance process is liable to various forms of abuse. Despite these considerations, we have placed the entirety of the burden of service authenticity and session encryption onto TLS, to the point that other related efforts, such as IPsec, BGP routing security, and *Domain Name System Security Extensions* (DNSSEC) in the DNS, are generally perceived as optional extras rather than basic essentials to be included the security toolkit.

Applications and Services

This layer has also seen quite profound changes over the past quarter century, tracking the progress of increasing technical capability as well as consumer demands. In the late 1990s, the Internet was on the cusp of portal mania, where *LookSmart* was the darling of the Internet boom and everyone was trying to promote their own favourite "one stop shop" for all your Internet needs.

By 1998 the *AltaVista* search engine had made its debut, and these content-collation portals were already becoming passé. This change, from compiling directories and lists to active search, completely changed the Internet. These days we simply assume that we can type any query we want to into a search engine and the search machinery will deliver a set of pointers to relevant documents. And every time it occurs our expectations about the quality and utility of search engines are reinforced. Content is also changing as a result, as users no longer remain on a *site* and navigate around the site. Instead, users are driving the search engines, and pulling the relevant pages without reference to any other material. But it has not stopped there. Search engines are morphing into "instant answer machines," where instead of providing a set of pointers to sources where there is a high level of correlation between the source and the question, the search engine attempts to extract material from the source and show what it believes is the answer to the implicit question in the search term. Even this process is just a way point in a longer journey, and today we are seeing *Artificial Intelligence* (AI) chat bots appearing, where the underlying data set that has been indexed by the search machinery is now being used as a corpus of data to drive an AI chat engine. The interaction is based on a natural language model.

If you thought of the Internet as an information resource, then the use of AI in this manner is a disturbing step. In this AI model the responding system generates plausible, but very definitely not necessarily factual, natural language responses to the implicit question in the query.

It's challenging to see this path from indexing data sources and matching query terms to the terms that primary sources use to one of a natural language generator that produces textual responses that are not grounded in facts, nor necessarily derived from primary sources, as being progress! Despite such misgivings about the deliberate abasement of the quality of the Internet as an information resource, this shift does fit into a larger picture of the transformation of the Internet to a mass entertainment vehicle, which is much of the driving force in today's content world.

Social Media

A related area of profound change has been the rise of social media. The television, radio, film, and print industries had evolved to use content mediators, compilers, and editors to curate their content, and the widespread deployment of highly capable user devices allowed end users to directly perform content production without the need to engage with mediators or producers. This situation has transformed many societies, and the social media platforms, including YouTube, Flickr, Facebook, Instagram, and TikTok, have been rocketed into societal prominence, prompting major debates about the role of these platforms and levels of societal influence that such platforms can generate.

Underlying these changes is another significant development, namely the change in the content economy. In 1998 content providers and ISPs were eyeing each other in an effort to gain user revenue. Content providers were unable to make pay-per-view and other forms of direct financial relationships with users work in their favour and argued that ISPs should fund content. After all, they pointed out, the only reason users paid for Internet access was the perceived value of the content they found there. ISPs, on the other hand, insisted that content providers were enjoying a "free ride" across the ISP-funded infrastructure, and content providers should contribute to network costs. The model that has gained ascendency as a result of this unresolved tension is that of advertisement-funded content services, and this model has been able to sustain a vastly richer, larger, and more compelling content environment.

However, using this model comes at a price, and in this case the price lies in the motivations of the platforms that perform ad delivery. The critical objective now is to engage the user for longer periods, so that they can present more ads and glean more information about the user's profile. Merely informing a user is largely a transactional interaction, whereas entertaining a user can be far more lucrative in terms of generating advertising revenue because of the longer attention span. This model has been highly successful for some content players, particularly the current giants of streaming content, and it's therefore unsurprising that the largest entities in the content world, such as Alphabet, Microsoft, Amazon, and Apple, are more valuable in terms of market capitalization than their counterparts in the carriage world. We are now seeing the next round of the friction between content and carriage, where the access network operators are arguing that the content players should contribute to the costs of access carriage.

The *Domain Name System* (DNS) also merits a mention in this section. From one perspective, little has changed in this space, and the DNS name-resolution protocol hasn't changed to any appreciable extent. In some sense that's true, but at the same time there have been some significant changes.

DNS

The first of these changes is adoption of *Domain Name System Security Extensions* (DNSSEC), a framework that allows DNS clients to validate the answers that they receive from the DNS. The DNS has always been a point of security vulnerability in the Internet in that it has always been prone to various forms of attack where false answers are substituted in place of the genuine answer. DNSSEC provides a digital signature record to each normal record, and also implements an interlocked chain of signatures to link to the key associated with the root zone. A client may request the signature record to be provided with the normal response, and then make further requests to construct the validation chain all the way to the root zone. Successful validation assures a client that the data provided in the original response is authentic and current. The root zone of the DNS was first signed in 2010, but adoption of DNSSEC has been slow. While the addition of such a validation mechanism is undoubtedly a step forward in protecting users against various forms of name-based interference, the cost is increased fragility of the DNS and increased resolution times. One underlying problem is that the addition of digital signatures to a DNS response is highly likely to push the DNS into sending large responses, and large responses over a UDP-based transport is prone to fragmentation-based unreliability, and the switch to use TCP also takes time. What this problem has implied is that the path to adoption of DNSSEC has been slow, despite the obvious protections it can provide regarding potential tampering with the DNS.

The second major theme of change in the DNS concerns the larger issue of pervasive monitoring in the DNS, highlighted by the Snowden revelations of 2013. Most Internet transactions start with a call to the DNS, and the meta-data contained in DNS queries and responses provides a rich real-time profile of user activity, both in general and potentially on a user-by-user basis. This situation has prompted a concerted effort to improve the privacy aspects of the DNS as a protocol. One approach has been to take the existing use of DNS across a TCP session and add TLS to the TCP session, so the contents of the interaction between the client and the DNS server are impervious to third-party inspection or manipulation. This approach can be taken a step further with DNS over *Hypertext Transfer Protocol Secure* (HTTPS)/2, where the DNS payload has a lightweight HTTP wrapper in addition to TLS. This approach allows DNS traffic to be melded in with all other HTTP traffic as a further step of obscuring DNS transactions. More recently we have seen DNS over QUIC, using QUIC faster session start times and fast open capabilities to improve the performance of the arrangement, and DNS over HTTPS/3, which combines QUIC with HTTP object semantics.

The primary focus of this work has been the part of the DNS where the client's stub resolver interacts with a recursive resolver, because this scenario identifies the client. The useful property of this part of the DNS is that the same client/server setup is used repeatedly, so either a long-held secure transport session or a fast-reopen session can amortise the high setup cost of a reliable secure session over many subsequent queries, making the overall cost of such a secure transport arrangement more palatable.

Such measures still have some security problems, as the recursive resolver is privy to both the client's identity and the DNS queries that they make. Recent work has begun on an "oblivious" model of DNS operation, where the recursive resolver function is split in two and two layers of encryption are used. The client talks to the first party, a DNS relay over an encrypted session, and passes it a query that has been encrypted using the public key of the second party, the recursive resolver. The relay resends the encrypted DNS query to the recursive resolver to resolve. The first party knows the identity of the client, but not the DNS query that is being made. The second party knows the DNS query, but not the identity of the client.

This work on DNS privacy has extended into the scenarios of the recursive resolver talking with authoritative name servers, although it's unclear as to the extent of the security benefits (because the end user is not identified directly in such queries), nor is session reuse as feasible in this scenario.

Cloud

In many ways applications and services have been the high frontier of innovation in the Internet in this period. An entire revolution in open interconnection of content elements has taken place, and content is now a very malleable concept. It is no longer the case of "my computer, my applications, my workspace" or "your server, your content" but an emerging model where not only the workspace for each user is held in the network, but where the applications and services themselves are part of the network, and all are accessed through a generic mechanism based around permutations of the HTTPS access model. This world is one of the so-called *Cloud Services*. The world of cloud services takes advantage of abundance in computation, storage, and communications resources, and rather than a network facilitating users to connect to service delivery points, the cloud model inverts the model and attempts to bring replicant copies of content and services closer to the user. If distance equates to cost and performance in the networking world, then the cloud model dramatically shortens the distance between consumer and content, with obvious implications in terms of cost and performance reductions. The clouded Internet can achieve extremely challenging performance and cost objectives by changing the provisioning model of the content and service from "just-in-time" on-demand service to "just-in-case" pre-provisioning of local caches so that the local cache is ready if a local client accesses the service.

Cyber Hostility

We still are under relentless attack at all levels. We are beset by data leaks, surveillance, profiling, disruption, and extortion.

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, such as UDP reflection attacks where a single 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 needs to be done. A small query stream can result in a massive attack. UDP protocols such as SNMP, the *Network Time Protocol* (NTP), the DNS, and *memcached* have been used in the past and doubtless will be used again.

Why can't we fix this problem? 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 was published more than two decades ago, in 2000. Yet massive UDP-based attacks with forged source addresses still persist today. Aged computer systems with known vulnerabilities continue to be connected to the Internet and are readily transformed into attack bots.

The picture of attacks is also becoming more ominous. Although we previously attributed these hostile attacks to "hackers," we quickly realised that a significant component of them 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 vulnerability exploitation that are considered 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 entirely 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 attacks is not getting any better. Indeed, it's getting 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 withstand attack. These days only a handful of platforms can offer resilient services, and even then, it's unclear whether they could withstand the most extreme of attacks.

A constant background level of scanning and probing goes 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 to 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 is a sobering thought 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.

IoT

What makes this scenario even more depressing is the portent of the so-called *Internet of Things* (IoT). 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, handheld devices, and now wrist computers.

Where next? In the vision of the IoT, we are going to expand the Internet beyond people and press on 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 affect others in potentially malicious ways.

If such devices were constantly inspected and managed, we might see evidence of aberrant behaviour and correct it. But these devices are unmanaged and all but invisible. Examples include the controller for a web camera, the so-called "smart" thing in a smart television, or the controls for anything from a washing machine to a goods locomotive. Nobody is looking after these devices. When we think of an IoT we think of a world of weather stations, webcams, "smart" cars, personal fitness monitors, and similar things.

But what we tend to forget is that all of these devices are built on 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 it's 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" device. It would be nice to think that we have 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 IoT will continue to be a marketplace where the compromises between price and quality will continue to push us on to the side of cheap rather than secure. What is 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? So far, we have not found workable answers to this question.

Our ability to effectively defend the network and its connected hosts continues to be, on the whole, ineffectual. Anyone who still has trust in the integrity of the systems that make up the digital world is just hopelessly naive. This space is toxic and hostile, and we still have no idea how we can shift it to a different state that can resist such erosive and insidious attacks. But somehow, we are evidently not deterred by all this information. Somehow each of us has found a way to make the Internet work for us.

The Business of the Internet

As much as the application environment of the Internet has been on a wild ride over the past 25 years, the business environment has also had its tickets on the same roller coaster ride, and the list of business winners and losers includes some of the historical giants of the telephone world as well as the Internet-bred new wave of entrants.

In 1998, despite the growing momentum of public awareness, the Internet was still largely a curiosity. Its environment was inhabited by geeks, game players, and academics, whose rites of initiation were quite arcane. As a part of the data networking sector, the Internet was just one further activity among many, and the level of attention from the mainstream telco sector was still relatively low. Most Internet users were customers of independent ISPs, and the business relationship between the ISP sector and the telco was tense and acrimonious. The ISPs were seen as opportunistic leeches on the telco industry; they ordered large banks of phone lines, but never made any calls; their customers did not hang up after 3 minutes, but kept their calls open for hours or even days at a time, and they kept on ordering ever-larger inventories of transmission capacity, yet had business plans that made scribblings on the back of an envelope look professional by comparison.

The telco was unwilling to make large long-term capital investments in additional communications infrastructure to pander to the extravagant demands of a wildcat set of Internet speculators and their fellow travellers.

The telco, on the other hand, was slow, expensive, inconsistent, ill-informed, and hostile to the ISP business. The telco wanted financial settlements and bit-level accounting while the ISP industry appeared to manage quite well with a far simpler system of peering and tiering that avoided putting a value on individual packets or flows.

This relationship was never going to last, and it resolved itself in ways that in retrospect were quite predictable. From the telco perspective, it quickly became apparent that the only reasons the telco was being pushed to install additional network capacity at ever-increasing rates were demands from the ISP sector. From the ISP perspective, the only way to grow at a rate that matched customer demand was to become one's own carrier and take over infrastructure investment. And, in various ways, both outcomes occurred. Telcos bought up ISPs, and ISPs became infrastructure carriers.

All this activity generated considerable investor interest, and the rapid value escalation of the ISP industry and then the entire Internet sector generated the levels of wild-eyed optimism that are associated only with an exceptional boom. By 2000 almost anything associated with the Internet, whether it was a simple portal, a new browser development, a search engine, or an ISP, attracted investor attention, and the valuations of Internet start-ups achieved dizzying heights. Of course, one of the basic lessons of economic history is that every boom has an ensuing bust, and in 2001 the Internet collapse happened. The bust was as inevitable and as brutal as the preceding boom was euphoric. But, like the railway boom and bust of the 1840s, after the wreckage was cleared away what remained was a viable, and indeed a valuable, industry.

By 2003 the era of the independent retail ISP was effectively over. But it reshaped itself dramatically with the introduction of mobile services. It was the old telco sector that had secured spectrum allocations in the bidding wars in the early 2000s, and while they had thought that mobile voice would be the reason why these investments would make sense, it was mobile Internet services that proved to be the lasting service model. In this period, the Internet was the amalgam of the largest of the original ISP, the transformed cable television operators, and the mobile providers. Each national regime was populated with some three to five major service providers, and the business started to stabilise around this model.

Into this world came the content world of the Internet, using cloud-based models of service delivery to circumvent communications bottlenecks in the long-haul transit sector. They embarked on service models that included advertiser-funded models of content generation and delivery and direct subscription models, and the result has been so effective that the value of this sector is far greater than the traditional ISP and carriage sector. The content world is now the major funder of subsea cable systems, and the carriage world has been very reluctantly pushed into an undistinguished commodity role as a result.

This situation is reflective of a broader process of technology permeation. The telephone world used the network as the major focus of technology and investment. The edge devices, telephone handsets, were simple, cheap devices, whereas network switches and transmission elements were built to exacting and expensive standards. As we attached computers to the edges of the network, these devices were able to tolerate a broader spectrum of network behaviours, and had a lower base-level expectation of behaviour. Consequently, value has moved out from the core of the network to its edges.

But this process has also been reflected within these edge devices. We started with a model of a highly capable and complicated operating system platform, and relatively simple applications that used platform services. Some 25 years ago the release of Windows 98 was a Big Thing, and rightly so. As these edge devices become more capable and have higher processing capability, more local storage applications have elected to take on more of the responsibility in terms of the user's experience. In doing so they no longer rely on the release schedules of the platform provider, and they are no longer as concerned about the level of control being exercised by this platform provider and gaining an essential level of self-control. Modern browsers (Chrome and a few far smaller fellow travellers) are far more complex than most operating systems, and they continue to subsume functions and roles that the platform previously carried out. With DNS over HTTPS, the task of DNS name resolution can be transformed to an application function, rather than a common platform function. With QUIC, the transport protocol itself has been subsumed into the application space.

Not only have we seen the commoditisation of the network over the past 25 years, we have also seen similar commoditisation pressures on the end-device platforms and on the operating systems used on these devices. Even the browser space has been commoditised. The brunt of competitive differentiation in this industry has been pushed up the protocol stack into the content and service economy, and there is the distinct feeling that even in that space competitive differentiation is perhaps a misnomer, and what we have is a synthetic form of competition between a select small group of digital service-delivery behemoths that in any other time and context would probably be called a cartel.

What Now?

It's been a revolutionary quarter-century for us all, and the Internet has directly or indirectly touched the lives of almost every person on this planet. Current estimates put the number of regular Internet users at one half of the world's population.

Over this period, some of our expectations were achieved and then surpassed with apparent ease, while others remained elusive. And some things occurred that were entirely unanticipated. At the same time, very little of the Internet we have today was confidently predicted in 1998, while many of the problems we saw in 1998 remain problems today.

This work-in-progress means the next quarter-century will probably see the same level of intensity of yet more structural changes to the global communications sector. And that is a somewhat scary prospect, given the collection of other challenges that we will all confront in the coming decades. At the same time, I think it would be good to believe that the debut of the Internet in our world has completely rewritten what it means to communicate, the way in which we can share our experience and knowledge, and, hopefully, the ways in which we can work together on these challenges.

References and Further Reading

The Internet Protocol Journal has published articles on all the major aspects of the technical evolution of the Internet over the past 25 years. To illustrate the extraordinary breadth of these articles, I have included as references here some pointers to articles that have been published in IPJ.

[1] William Stallings, "SSL: Foundation for Web Security," *The Internet Protocol Journal*, Volume 1, No. 1, June 1998.

[2] Fred Avolio, "Firewalls and Internet Security," *The Internet Protocol Journal*, Volume 2, No. 2, June 1999.

[3] William Stallings, "Gigabit Ethernet," *The Internet Protocol Journal*, Volume 2, No. 3, September 1999.

[4] Mark Handley and Jon Crowcroft, "Internet Multicast Today," Volume 2, No. 4, December 1999.

[5] Geoff Huston, "Quality of Service – Fact or Fiction?," *The Internet Protocol Journal*, Volume 3, No. 1, March 2000.

[6] Geoff Huston, "The Future for TCP," *The Internet Protocol Journal*, Volume 3, No. 3, September 2000.

[7] Chris Lonvick, "Securing the Infrastructure," *The Internet Protocol Journal*, Volume 3, No. 3, September 2000.

[8] William Stallings, "Mobile IP," *The Internet Protocol Journal*, Volume 4, No. 2, June 2001.

[9] Geoff Huston, "The Middleware Muddle," *The Internet Protocol Journal*, Volume 4, No. 2, June 2001.

[10] William Stallings, "MPLS," *The Internet Protocol Journal*, Volume 4, No. 3, September 2001.

[11] Stephen Kent, "Securing BGP: S-BGP," *The Internet Protocol Journal*, Volume 6, No. 3, September 2003.

[12] Russ White, "Securing BGP: soBGP," *The Internet Protocol Journal*, Volume 6, No. 3, September 2003.

[13] Geoff Huston, "Anatomy: Inside Network Address Translators," *The Internet Protocol Journal*, Volume 7, No. 3, September 2004

[14] Daniel McCarney, "Automatic Certificate Management," *The Internet Protocol Journal*, Volume 20, No. 2, June 2017.

[15] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki, "Distributed Denial of Service Attacks," *The Internet Protocol Journal*, Volume 7, No. 4, December 2004.

[16] David Crocker, "Challenges in Anti-Spam Efforts," *The Internet Protocol Journal*, Volume 8, No. 4, December 2005.

[17] Vint Cerf, "A Decade of Internet Evolution," *The Internet Protocol Journal*, Volume 11, No 2, June 2008.

[18] Geoff Huston, "A Decade in the Life of the Internet," *The Internet Protocol Journa*l, Volume 11, No. 2, June 2008

[19] Thayumanavan Sridhar, "Cloud Computing – A Primer," *The Internet Protocol Journal*, Volume 12, No. 3, September 2009, Vol. 12, No. 4, December 2009.

[20] Bob Hinden, "The Internet of Insecure Things," *The Internet Protocol Journal*, Volume 20, No. 1, March 2017.

[21] Andrei Robachevsky, "Improving Routing Security," *The Internet Protocol Journal*, Volume 22, No. 2, July 2019.

[22] Geoff Huston, "DNS Privacy," *The Internet Protocol Journal*, Volume 22, No. 2, July 2019.

[23] David Strom, "So You Want to Sell Your IPv4 Address Block?," *The Internet Protocol Journal*, Volume 23, No. 2, September 2020.

[24] Geoff Huston, "DNS Trends," *The Internet Protocol Journal*, Volume 24, No. 1, March 2021.

[25] Geoff Huston, "Securing Inter-Domain Routing," *The Internet Protocol Journal*, Volume 24, No. 3, October 2021, and Volume 25, No. 1, April 2022

[26] Geoff Huston, "Comparing TCP and QUIC," *The Internet Protocol Journal*, Volume 25, No. 3, December 2022.

[27] Geoff Huston, "Protocol Basics: The Network Time Protocol," *The Internet Protocol Journal*, Volume 15, No. 4, December 2012.

[28] Burton S. Kaliski Jr., "Minimized DNS Resolution: Into the Penumbra," *The Internet Protocol Journal*, Volume 25, No. 3, December 2022.

GEOFF HUSTON AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990s. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001. At various times Geoff has worked as an Internet researcher, an ISP systems architect, and a network operator. E-mail: **gih@apnic.net**

Supporters and Sponsors

Supporters

Internet Society

CISCO

Diamond Sponsors

Your logo here!

Ruby Sponsors

ICANN

Sapphire Sponsors

APNIC FOUNDATION

Emerald Sponsors

Akamai

amsix

APRICOT Asia Pacific Regional Internet Conference on Operational Technologies

DE-CIX

EQUINIX

Google

identity digital

JPRS

JUNIPER NETWORKS

lacnic

LinkedIn

linx

netskope

NSRC Network Startup Resource Center

NTT Communications

RIPE NCC RIPE NETWORK COORDINATION CENTRE

TEAM CYMRU

VERISIGN

WIDE PROJECT

Corporate Subscriptions

AFRINIC The Internet Numbers Registry for Africa

COMCAST

DNS-OARC

Edgio

IPXO

ISC Internet Systems Consortium

IWL

qa cafe

For more information about sponsorship, please contact **sponsor@protocoljournal.org**

Thank You!

Publication of IPJ is made possible by organizations and individuals around the world dedicated to the design, growth, evolution, and operation of the global Internet and private networks built on the Internet Protocol. The following individuals have provided support to IPJ. You can join them by visiting **http://tinyurl.com/IPJ-donate**

Kjetil Aas
Fabrizio Accatino
Michael Achola
Martin Adkins
Melchior Aelmans
Christopher Affleck
Scott Aitken
Jacobus Akkerhuis
Antonio Cuñat Alario
William Allaire
Nicola Altan
Shane Amante
Marcelo do Amaral
Matteo D'Ambrosio
Selva Anandavel
Jens Andersson
Danish Ansari
Finn Arildsen
Tim Armstrong
Richard Artes
Michael Aschwanden
David Atkins
Jac Backus
Jaime Badua
Bent Bagger
Eric Baker
Fred Baker
Santosh Balagopalan
William Baltas
David Bandinelli
A C Barber
Benjamin Barkin-Wilkins
Feras Batainah
Michael Bazarewsky
David Belson
Richard Bennett
Matthew Best
Hidde Beumer
Pier Paolo Biagi
Arturo Bianchi
John Bigrow
Orvar Ari Bjarnason
Tyson Blanchard
Axel Boeger
Keith Bogart
Mirko Bonadei
Roberto Bonalumi
Lolke Boonstra
Julie Bottorff Photography
Gerry Boudreaux
Leen de Braal
Kevin Breit
Thomas Bridge
Ilia Bromberg
Lukasz Bromirski
Václav Brožík
Christophe Brun
Gareth Bryan
Ron Buchalski
Paul Buchanan
Stefan Buckmann
Caner Budakoglu
Darrell Budic
BugWorks
Scott Burleigh
Chad Burnham
Randy Bush
Colin Butcher
Jon Harald Bøvre
Olivier Cahagne
Antoine Camerlo
Tracy Camp
Brian Candler
Fabio Caneparo
Roberto Canonico
David Cardwell
Richard Carrara
John Cavanaugh
Lj Cemeras
Dave Chapman
Stefanos Charchalakis
Molly Cheam
Greg Chisholm
David Chosrova
Marcin Cieslak
Lauris Cikovskis
Brad Clark
Narelle Clark
Horst Clausen
James Cliver
Guido Coenders
Joseph Connolly
Steve Corbató
Brian Courtney
Beth and Steve Crocker
Dave Crocker
Kevin Croes
John Curran
André Danthine
Morgan Davis
Jeff Day
Rodolfo Delgado-Bueno
Julien Dhallenne
Freek Dijkstra
Geert Van Dijk
David Dillow
Richard Dodsworth
Ernesto Doelling
Michael Dolan
Eugene Doroniuk
Michael Dragone
Joshua Dreier
Lutz Drink
Aaron Dudek
Dmitriy Dudko
Andrew Dul
Joan Marc Riera Duocastella
Pedro Duque
Holger Durer
Karlheinz Dölger
Mark Eanes
Andrew Edwards
Peter Robert Egli
George Ehlers
Peter Eisses
Torbjörn Eklöv
Y Ertur
ERNW GmbH
ESdatCo
Steve Esquivel
Jay Etchings
Mikhail Evstiounin
Bill Fenner
Paul Ferguson
Ricardo Ferreira
Kent Fichtner
Armin Fisslthaler
Michael Fiumano
The Flirble Organisation
Jean-Pierre Forcioli
Gary Ford
Susan Forney
Christopher Forsyth
Andrew Fox
Craig Fox
Fausto Franceschini
Valerie Fronczak
Tomislav Futivic
Laurence Gagliani
Edward Gallagher
Andrew Gallo
Chris Gamboni
Xosé Bravo Garcia
Osvaldo Gazzaniga
Kevin Gee
Greg Giessow
John Gilbert
Serge Van Ginderachter
Greg Goddard
Tiago Goncalves
Ron Goodheart
Octavio Alfageme Gorostiaga
Barry Greene
Jeffrey Greene
Richard Gregor
Martijn Groenleer
Geert Jan de Groot
Ólafur Guðmundsson
Christopher Guemez
Gulf Coast Shots
Sheryll de Guzman
Rex Hale
Jason Hall
James Hamilton
Darow Han
Handy Networks LLC
Stephen Hanna
Martin Hannigan
John Hardin
David Harper
Edward Hauser
David Hauweele
Marilyn Hay
Headcrafts SRLS
Hidde van der Heide
Johan Helsingius
Robert Hinden
Damien Holloway
Alain Van Hoof
Edward Hotard
Bill Huber
Hagen Hultzsch
Kauto Huopio
Asbjørn Højmark
Kevin Iddles
Mika Ilvesmaki
Karsten Iwen
Joseph Jackson
David Jaffe
Ashford Jaggernauth
Thomas Jalkanen
Jozef Janitor
Martijn Jansen
John Jarvis
Dennis Jennings
Edward Jennings
Aart Jochem
Nils Johansson
Brian Johnson
Curtis Johnson
Richard Johnson
Jim Johnston
Jonatan Jonasson
Daniel Jones
Gary Jones
Jerry Jones
Michael Jones
Amar Joshi
Javier Juan
David Jump
Anders Marius Jørgensen
Merike Kaeo
Andrew Kaiser
Naoki Kambe
Christos Karayiannis
Daniel Karrenberg
David Kekar
Stuart Kendrick
Robert Kent
Jithin Kesavan
Jubal Kessler
Shan Ali Khan
Nabeel Khatri
Dae Young Kim
William W. H. Kimandu
John King
Russell Kirk
Gary Klesk
Anthony Klopp
Henry Kluge
Michael Kluk
Andrew Koch
Ia Kochiashvili
Carsten Koempe
Richard Koene
Alexader Kogan
Matthijs Koot
Antonin Kral
Robert Krejčí
John Kristoff
Terje Krogdahl
Bobby Krupczak
Murray Kucherawy
Warren Kumari
George Kuo
Dirk Kurfuerst
Mathias Körber
Darrell Lack
Andrew Lamb
Richard Lamb
Yan Landriault
Edwin Lang
Sig Lange
Markus Langenmair
Fred Langham
Tracy LaQuey Parker
Alex Latzko
Jose Antonio Lazaro Lazaro
Antonio Leding
Rick van Leeuwen

Simon Leinen
Robert Lewis
Christian Liberale
Martin Lillepuu
Roger Lindholm
Link Light Networks
Chris and Janet Lonvick
Sergio Loreti
Eric Louie
Adam Loveless
Josh Lowe
Guillermo a Loyola
Hannes Lubich
Dan Lynch
David MacDuffie
Sanya Madan
Miroslav Madić
Alexis Madriz
Carl Malamud
Jonathan Maldonado
Michael Malik
Tarmo Mamers
Yogesh Mangar
John Mann
Bill Manning
Harold March
Vincent Marchand
Normando Marcolongo
Gabriel Marroquin
David Martin
Jim Martin
Ruben Tripiana Martin
Timothy Martin
Carles Mateu
Juan Jose Marin Martinez
Ioan Maxim
David Mazel
Miles McCredie
Brian McCullough
Joe McEachern
Alexander McKenzie
Jay McMaster
Mark Mc Nicholas
Olaf Mehlberg
Carsten Melberg
Kevin Menezes
Bart Jan Menkveld
Sean Mentzer
Eduard Metz
William Mills
David Millsom
Desiree Miloshevic
Joost van der Minnen
Thomas Mino
Rob Minshall
Wijnand Modderman-Lenstra
Mohammad Moghaddas
Charles Monson
Andrea Montefusco
Fernando Montenegro
Roberto Montoya
Joel Moore
John More
Maurizio Moroni
Brian Mort
Soenke Mumm
Tariq Mustafa
Stuart Nadin
Michel Nakhla
Mazdak Rajabi Nasab
Krishna Natarajan
Naveen Nathan
Darryl Newman
Thomas Nikolajsen
Paul Nikolich
Travis Northrup
Marijana Novakovic
David Oates
Ovidiu Obersterescu
Jim Oplotnik
Tim O'Brien
Mike O'Connor
Mike O'Dell
John O'Neill
Carl Önne
Packet Consulting Limited
Carlos Astor Araujo Palmeira
Gordon Palmer
Alexis Panagopoulos
Gaurav Panwar
Chris Parker
Alex Parkinson
Craig Partridge
Manuel Uruena Pascual
Ricardo Patara
Dipesh Patel
Dan Paynter
Leif Eric Pedersen
Rui Sao Pedro
Juan Pena
Chris Perkins
Michael Petry
Alexander Peuchert
David Phelan
Harald Pilz
Derrell Piper
Rob Pirnie
Jorge Ivan Pincay Ponce
Marc Vives Piza
Victoria Poncini
Blahoslav Popela
Andrew Potter
Ian Potts
Eduard Llull Pou
Tim Pozar
David Raistrick
Priyan R Rajeevan
Balaji Rajendran
Paul Rathbone
William Rawlings
Mujtiba Raza Rizvi
Bill Reid
Petr Rejhon
Robert Remenyi
Rodrigo Ribeiro
Glenn Ricart
Justin Richards
Rafael Riera
Mark Risinger
Fernando Robayo
Michael Roberts
Gregory Robinson
Ron Rockrohr
Carlos Rodrigues
Magnus Romedahl
Lex Van Roon
Marshall Rose
Alessandra Rosi
David Ross
William Ross
Boudhayan Roychowdhury
Carlos Rubio
Rainer Rudigier
Timo Ruiter
RustedMusic
Babak Saberi
George Sadowsky
Scott Sandefur
Sachin Sapkal
Arturas Satkovskis
PS Saunders
Richard Savoy
John Sayer
Phil Scarr
Gianpaolo Scassellati
Elizabeth Scheid
Jeroen Van Ingen Schenau
Carsten Scherb
Ernest Schirmer
Benson Schliesser
Philip Schneck
James Schneider
Peter Schoo
Dan Schrenk
Richard Schultz
Timothy Schwab
Roger Schwartz
SeenThere
Scott Seifel
Paul Selkirk
Andre Serralheiro
Yury Shefer
Yaron Sheffer
Doron Shikmoni
Tj Shumway
Jeffrey Sicuranza
Thorsten Sideboard
Greipur Sigurdsson
Fillipe Cajaiba da Silva
Andrew Simmons
Pradeep Singh
Henry Sinnreich
Geoff Sisson
John Sisson
Helge Skrivervik
Terry Slattery
Darren Sleeth
Richard Smit
Bob Smith
Courtney Smith
Eric Smith
Mark Smith
Tim Sneddon
Craig Snell
Job Snijders
Ronald Solano
Asit Som
Ignacio Soto Campos
Evandro Sousa
Peter Spekreijse
Thayumanavan Sridhar
Paul Stancik
Ralf Stempfer
Matthew Stenberg
Martin Štěpánek
Adrian Stevens
Clinton Stevens
John Streck
Martin Streule
David Strom
Colin Strutt
Viktor Sudakov
Edward-W. Suor
Vincent Surillo
Terence Charles Sweetser
T2Group
Roman Tarasov
David Theese
Douglas Thompson
Kerry Thompson
Lorin J Thompson
Fabrizio Tivano
Peter Tomsu Fine Art Photography
Joseph Toste
Rey Tucker
Sandro Tumini
Angelo Turetta
Michael Turzanski
Phil Tweedie
Steve Ulrich
Unitek Engineering AG
John Urbanek
Martin Urwaleck
Betsy Vanderpool
Surendran Vangadasalam
Ramnath Vasudha
Randy Veasley
Philip Venables
Buddy Venne
Alejandro Vennera
Luca Ventura
Scott Vermillion
Tom Vest
Peter Villemoes
Vista Global Coaching & Consulting
Dario Vitali
Rüdiger Volk
Jeffrey Wagner
Don Wahl
Michael L Wahrman
Lakhinder Walia
Laurence Walker
Randy Watts
Andrew Webster
Jd Wegner
Tim Weil
Westmoreland Engineering Inc.
Rick Wesson
Peter Whimp
Russ White
Jurrien Wijlhuizen
Derick Winkworth
Pindar Wong
Makarand Yerawadekar
Phillip Yialeloglou
Janko Zavernik
Bernd Zeimetz
Muhammad Ziad Ziayuddin
Tom Zingale
Jose Zumalave
Romeo Zwart
廖明沂.

Follow us on Twitter and Facebook

@protocoljournal

https://www.facebook.com/newipj

The Internet Protocol Journal
Link Fulfillment
7650 Marathon Dr., Suite E
Livermore, CA 94550

CHANGE SERVICE REQUESTED

The Internet Protocol Journal is published quarterly and supported by the Internet Society and other organizations and individuals around the world dedicated to the design, growth, evolution, and operation of the global Internet and private networks built on the Internet Protocol.

Email: ipj@protocoljournal.org
Web: www.protocoljournal.org