The Internet Protocol Journal

December 2019 Volume 22, Number 3

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

"A major design feature of the *Internet Protocol* (IP) is its ability to run over a variety of underlying network technologies. If you look through the *Request For Comments* (RFC) document series, you will find numerous specifications of the form "IP over xxx," where "xxx" is anything from Ethernet to X.25, Frame Relay, Bluetooth, WiFi, and even "Avian Carriers" (pigeons), the latter being one of the more famous April Fools RFCs. Because each of these technologies has different capabilities in terms of how much data can be carried in a "packet" or datagram, IP employs the concept of *fragmentation* and *reassembly* in cases where the originating datagram is larger than what the underlying network medium can support."

That paragraph is a quote from our June 2016 issue (Volume 19, No. 2) in which Geoff Huston described various aspects of IPv4 and IPv6 fragmentation. In this issue he explains how the *Transmission Control Protocol* (TCP) and its concept of a *Maximum Segment Size* (MSS) might interact with IP fragmentation even if this interaction is technically a "layer violation." His article presents measurement data on TCP MSS handshakes recorded by APNIC in August 2019.

The Internet has its origins in the *Advanced Research Projects Agency Network* (ARPANET), which began operation just over 50 years ago, in October 1969, with only two nodes. We asked Vint Cerf, one of the "Fathers of the Internet," to reflect on this milestone.

As always, we welcome your feedback and suggestions on anything you read in this journal. Letters to the Editor may be edited for clarity and length and can be sent to **ipj@protocoljournal.org**. Please make sure your subscription details are accurate. In this issue you will also find a summary of our Privacy Policy.

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

MSS Values of TCP

by Geoff Huston, APNIC

The *Transmission Control Protocol* (TCP) *Maximum Segment Size* (MSS) has been under some examination in recent months because an operating system vulnerability related to the Linux implementation of TCP occurred; it is described in CVE-2019-11477, 11478, and 11479[1]. One of the effective workarounds to avoid this problem is to block all TCP connection attempts that use a MSS value of 500 or lower.

What is the impact of such a TCP filter policy? What is being used as MSS values? How will a drop filter of TCP sessions with an MSS value of 500 or lower affect the Internet user base? In the *Asia-Pacific Network Information Centre* (APNIC) measurement platform we have assembled a large collection of recorded TCP handshakes, each of which contains a record of the TCP MSS exchange. Let's look at the MSS settings.

The TCP MSS Parameter

The MSS parameter is a part of the *Options* field in the TCP initial handshake that specifies the largest amount of data that a TCP speaker can receive in a single TCP segment[2]. The MSS relates to the TCP input buffer size within the implementation as packets are passed from the IP module to the TCP module. Each direction of TCP traffic uses its own MSS value, as this value is receiver-specified. The two ends don't have to agree on a common value because it acts as a constraint on the sender to send TCP segments no larger than this MSS value. But of course smaller TCP segments can always be sent. This MSS value can vary between the forward and reverse directions of a TCP data flow.

The MSS value does not count the TCP header or the IP header. The received IP datagram containing a TCP segment may be self-contained within a single packet, or it may be reconstructed from several fragmented pieces.

Because IP packet fragmentation is an IP-level issue, TCP should not directly concern itself with IP fragmentation in any case. In theory. In practice, a judicious setting of TCP MSS sizes that attempts to avoid sending TCP packets that incur IP-level packet fragmentation should be avoided!

Conventionally, the platform rather than the application sets the MSS value for a connection and the setting is applied to all TCP connections. But many operating system platforms provide a hook in the connection *Application Programming Interface* (API) for an application to specify the MSS value for a connection (such as the TCP_MAXSEG socket option).

What Is a "Good" MSS Value?

Getting the MSS value "just right" is important. While in theory the IP and TCP layers are largely independent, the practical reality is quite the opposite. Too high a value can lead to inefficient and even wedged TCP sessions due to issues with mishandling of IP fragmentation. The problem is that the sender may perform TCP segmentation by using the received MSS value as its guide and the resultant TCP packet is then far larger than the outgoing IP interface *Maximum Transmission Unit* (MTU) size, entailing the sender to perform IP-level fragmentation on the TCP packet.

It's also worth remembering that many of the TCP congestion control protocols use a rate acceleration based on an increase in the sending rate of 1 MSS of data per round-trip-time interval. Larger MSS values imply a faster rate of acceleration in such protocols, while smaller MSS values will lead to inefficiencies and may stall the sender, potentially leading to some congestion issues within the sender. The IPv4 packet contains a 16-bit packet identification number, implying that in order to avoid fragmentation reassembly issues not more than 65,536 IP packets should be in flight at any point in time.

Therefore, the combination of very small MSS values, long-held TCP sessions, and long-delay bandwidth network paths is certainly inefficient, but it should not necessarily represent any form of attack vector if the implementation of TCP is suitably robust. The recent security notices point to some platform vulnerabilities within the sender that are exposed by low MSS values.

What guidance is there in the RFCs on setting the TCP MSS value?

RFC 791[3] provides IP MTU guidance, stating that:

> "All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams."

Given that IP has no explicit MTU signalling capability, this explicit recommendation of obtaining assurance of the receiver's preparedness to accept larger IP datagrams presumably refers to the TCP MSS value.

RFC 879[4] provided quite explicit guidance about the TCP MSS value:

> "THE TCP MAXIMUM SEGMENT SIZE IS THE IP MAXIMUM DATAGRAM SIZE MINUS FORTY. The default IP Maximum Datagram Size is 576. The default TCP Maximum Segment Size is 536."

These documents were written prior to the specification of IPv6 of course, and in RFC 2460[5] the following guidance was given for IPv6:

> "When using TCP over IPv6, the MSS must be computed as the maximum packet size minus 60 octets."

It also states that:

> "IPv6 requires that every link in the internet have an MTU of 1280 octets or greater."

Taken together, RFC 2460 asserts that for TCP over IPv6 the MSS value would be expected to be 1,220 or greater.

These days the now-ancient Ethernet packet-framing specification still dominates the networking environment (although the old thick yellow coaxial cables and even the *Carrier Sense Multiple Access/Collision Detection* [CSMA/CD] 10-Mbps common bus protocol were both consigned to the networking section of silicon heaven years ago!). Thus the most common IP packet MTU is 1,500 octets.

Further clarification was provided in RFC 6691[6], "TCP Options and Maximum Segment Size," (published in July 2012):

> "When calculating the value to put in the TCP MSS option, the MTU value SHOULD be decreased by only the size of the fixed IP and TCP headers and SHOULD NOT be decreased to account for any possible IP or TCP options; conversely, the sender MUST reduce the TCP data length to account for any IP or TCP options that it is including in the packets that it sends. […] the goal is to avoid IP-level fragmentation of TCP packets."

That information implies that the most common anticipated TCP MSS values would correspond to a 1,500-octet MTU in both IPv4 and IPv6, further implying that we should see a MSS value of 1,460 in IPv4 and 1,440 in IPv6.

How well does practice line up with the theory?

Measuring TCP MSS Values

We looked at the MSS sizes in the HTTP(S) sessions offered by clients who connected to servers with our measurement as part of our large measurement program into IPv6 deployment. We collected all TCP handshakes that occurred in August 2019 and recorded the MSS values from the SYN packets received from the client systems.

We saw some 3B TCP sessions over this period, and after we removed the duplicate entries for multiple TCP sessions from the same endpoint within a similar timeframe with a common MSS value, we were left with some 551M unique TCP sessions.

Surprisingly enough, 202 endpoints offered an MSS value of 0, and 284 endpoints offered a value of 1. To put this data into perspective, this count of 486 endpoints represents 0.0001% (slightly less than 1 per million) of all observed TCP sessions. Small MSS values exist, but they are very much a rarity in the larger population of the Internet. A total of 20,488 sessions were opened with MSS values of 500 or lower (0.004%).

At the other end of the range of observed MSS values, three sessions used a value of 65,516 (the IPv4 maximum MTU minus 40). If we categorise any MSS value greater than 1,460 as some form of jumbo MSS, then we observed that 68,278 sessions used jumbo MSS values, or 0.012% of all TCP sessions.

As a side note, the network industry has never reached a clear agreement on exactly what a *jumbo frame* size should be. A value of 9,216 octets has been commonly quoted, as has the Internet2-defined value of 9,000 octets. The lack of agreement within the *Institute of Electrical and Electronics Engineers* (IEEE) on a single definition of a jumbo frame is not entirely unique, as many media-level protocols have used what could only be described in retrospect as idiosyncratic maximum MTU values. IEEE 802.5 *Token Ring* used an MTU of up to 4,464 octets, *Fiber Distributed Data Interface* (FDDI) used 4,532 octets, and IEEE 802.11 used 7,935 octets. Perhaps this diversity in media-based MTU values is not all that surprising, and what is perhaps more surprising is a current rough consensus of a commonly assumed MTU of 1,500 octets in the Internet, irrespective of the underlying media capabilities.

Table 1 shows the most common observed MSS values.

Table 1: Most Common MSS Values

Rank	MSS	Ratio	Rank	MSS	Ratio
1	1,460	17.6%	14	1,390	0.8%
2	1,400	16.4%	15	1,358	0.7%
3	1,370	11.3%	16	1,368	0.6%
4	1,452	8.7%	17	1,388	0.6%
5	1,440	8.3%	18	1,350	0.5%
6	1,360	6.9%	19	1,394	0.4%
7	1,412	5.1%	20	1,312	0.4%
8	1,300	4.3%	21	1,220	0.4%
9	1,380	3.7%	22	1,362	0.3%
10	1,420	3.5%	23	1,240	0.3%
11	1,432	1.8%	24	1,414	0.3%
12	1,410	1.3%	25	1,344	0.3%
13	1,340	1.3%			

The 1,460 value appears to correlate with a 1,500-octet MTU and a 40-octet IPv4 and TCP packet header. If the MSS value is calculated from the interface MTU size less the size of the IP and TCP headers, then we would expect the IPv6 MSS sizes to be 20 bytes less than the IPv4 MSS sizes.

We can separate the TCP MSS values used in IPv4 and IPv6, as shown in Table 2.

Table 2: Most Common MSS Values in IPv4 and IPv6

IPv4			IPv6	
Rank	MSS	Ratio	MSS	Ratio
1	1,460	17.6%	1,370	28.1%
2	1,400	16.4%	1,440	20.5%
3	1,370	11.3%	1,432	7.8%
4	1,452	8.7%	1,300	6.2%
5	1,440	8.3%	1,400	5.2%
6	1,360	6.9%	1,380	4.7%
7	1,412	5.1%	1,340	4.2%
8	1,300	4.3%	1,412	2.9%
9	1,380	3.7%	1,368	2.9%
10	1,420	3.5%	1,358	2.8%
11	1,432	1.8%	1,390	2.5%
12	1,410	1.3%	1,420	2.2%
13	1,340	1.3%	1,220	1.6%
14	1,390	0.8%	1,312	1.6%
15	1,358	0.7%	1,350	1.4%
16	1,368	0.6%	1,362	1.2%
17	1,388	0.6%	1,360	0.8%
18	1,350	0.5%	1,426	0.5%
19	1,394	0.4%	1,428	0.5%
20	1,312	0.4%	1,240	0.4%
21	1,220	0.4%	1,394	0.3%
22	1,362	0.3%	1,404	0.2%
23	1,240	0.3%	1,200	0.2%
24	1,414	0.3%	1,140	0.2%
25	1,344	0.3%	1,330	0.1%

The 1,370 value in IPv6 is somewhat unusual, as it corresponds to a MTU of 1,430 octets. It appears to be a common situation to use a 1,430-octet MTU in hosts, presumably as such a value (and any MTU value less than 1,460 octets) would minimise both the risks of IP fragmentation and path MTU issues that may arise from path element encapsulation that could be encountered when using IPv6.

If one takes the 1,460 MSS value in IPv4 and the 1,440 MSS value in IPv6 as an indicator of an underlying 1,500 MTU size, then relatively more endpoints are using a 1,500 MTU in IPv6 than in IPv4. (17.6% in IPv4 vs 20.5% in IPv6). In IPv6 there is a stronger consensus to use a single, smaller MSS value of 1,370 (28.1%) than there is in IPv4, where there is significant use of both 1,400 (16.4%) and 1,370 (11.3%) as MSS values.

The range of observed MSS values between 1,300 and 1,440 in both IPv4 and IPv6 points to the existence of a common action of constraining the IP MTU size in order to circumvent the possibility of IP fragmentation in both IPv4 and IPv6. I described the problem in 2009 in "A Tale of Two Protocols: IPv4, IPv6, MTUs and Fragmentation,"[7] and pointed out why a reduced MTU setting would be a reasonable response to this problem.

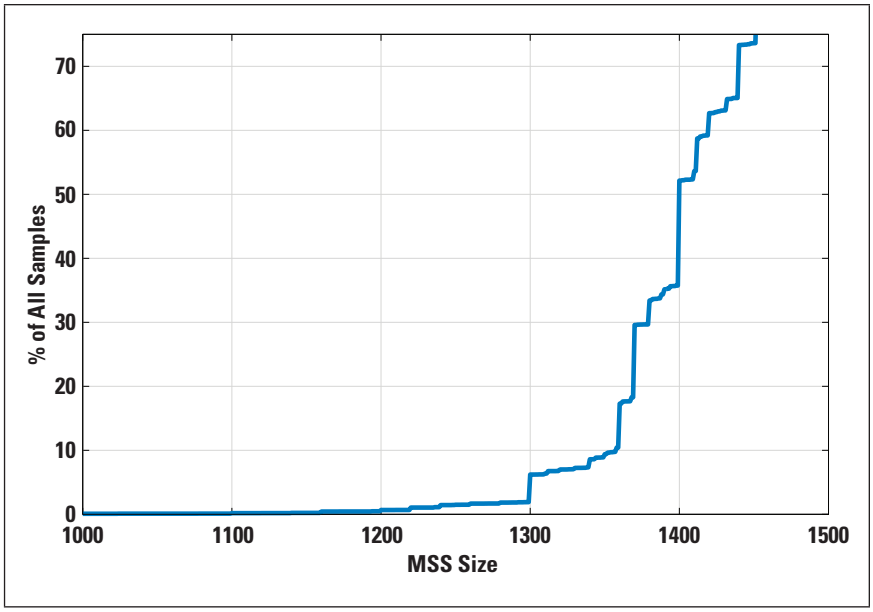
We also saw the MSS value of 536 in 38,359 cases, 38,245 of which were in IPv4 and 14 in IPv6, a value that appears to be derived from an assumed interface MTU of 576 octets and a 20-octet IPv4 packet header and 20-octet TCP header.

Oddly enough, the MSS value of 512 is more prevalent than that of 536, observed in 319,612 cases in IPv4 and not observed at all in IPv6.

While there is no particular media type that uses an MTU of 1,280 (which is the minimum unfragmented packet MTU size in IPv6), we had observed a minor clustering of MSS values at 1,220, with 1,892,966 IPv6 samples using this MSS value. Oddly enough, there were 7,268,197 cases of a 1,300-byte MSS value in IPv6.

Figure 1 on the following page shows the overall distribution of observed IPv4 MSS sizes.

Figure 1: Cumulative Distribution of TCP MSS Values



Incidence of Low MSS Values

Where might we see hosts with low (less than 500) MSS values?

In IPv4 almost half of all such systems are located in Germany and the Netherlands. Adding the data from systems offering small MSS values from France, South Korea, Bangladesh, Indonesia, Pakistan, and Brazil to the set encompasses some 90% of all hosts with MSS values less than 500. In the case of IPv6, more than half of the low MSS values are from hosts located in Germany, and 90% of all such hosts are located in Germany, Indonesia, the United States, Canada, Malaysia, and Brazil.

In terms of origin network in IPv4, the networks that contain the most hosts with observed low MSS values are operated by a large web-hosting enterprise. In IPv6 the majority of instances originate from a research centre in Germany. It may be that this high incidence of these very low MSS values in these networks could be due to some bug or operational misconfiguration in web-hosting equipment, or an unintentional configuration choice made by a client of this virtual system hosting service.

Incidence of High MSS Values

And where are hosts that use large TCP MSS values (values greater than 1,460)?

In IPv4 the United States contains 21% of all such hosts, and the somewhat diverse collection of India, Russia, and Ireland also each host some 4 to 6% of the total count of such hosts. The picture alters with IPv6, with half of all such hosts located in Japan and a little under one-quarter located in the United States.

At the network level the Amazon *Autonomous System Numbers* (ASNs) were most commonly found to be hosting high MSS-valued TCP stacks in IPv4, while the Japanese *Internet Service Providers* (ISPs) KDDI and NTT's OCN and Comcast in the United States were hosting high-valued MSS hosts in IPv6.

It appears that while some form of hosting or cloud system generates a large MSS value in IPv4, some form of configuration of ISP server product might be the cause of this behaviour in IPv6.

Incidence of Adjusted MSS Values

We can make the supposition that an MSS value of between 1,260 and 1,440 has been the result of a deliberate adjustment of the host MTU value in order to reduce the risk of packet fragmentation and path MTU black holes.

A path MTU black hole occurs when a server emits a packet that is too large for a network path element on the path to the receiver and, in the case of IPv4, either the *Don't Fragment* bit of the packet is set or the packet is an IPv6 packet, and the return path to the server is blocking *Internet Control Message Protocol* (ICMP) messages for some reason.

At this point the connection will stall. The sender is waiting for either an ACK of the data sequence number in the dropped packet or an ICMP packet to indicate that that there is an MTU problem. The ACK will never arrive as the packet has been dropped and the ICMP message has been blocked within the network.

The server will timeout and retransmit the large packet, to no effect. It may do so indefinitely unless some local overall session timeout is in effect, or TCP keepalives are in use.

The client has no outstanding data, so it will not retransmit and will just hang, waiting for a packet that will never arrive. TCP keepalives may identify this hung state and kill the hung TCP session.

We see these adjusted MSS values in those locations with a high IPv6 deployment volume—including India, the United States, Japan, and Vietnam.

What Values Should Be Used for TCP MSS?

A decade ago, the best advice around was to use a down-adjusted MSS value such as 1,300, 1,380, or even 1,400. The reason was to avoid path MTU issues, particularly when using IPv6, and the reason why path MTU issues were encountered in IPv6 was the prevalent use of IPv6-in-IPv4 encapsulation tunnels in IP transit paths and the widespread practice of firewall filtering of ICMPv6 *Packet Too Big* messages.

I'm not sure that ICMP filtering has improved or worsened in the last decade, but what has improved markedly is the use of "native" IPv6 transit paths in the Internet.

While it was probably foolhardy to use a 1,500 MTU and a 1,440 MSS with IPv6 a decade ago, it appears now to be not quite so foolhardy. Of course, not every tunnel has been removed and not every potential path MTU issue has been eliminated from the network, not every ICMPv6 filter has been removed, and not every fragment discard rule has been purged from firewalls, and they will probably never be completely purged. In relative terms the situation is better than it was a decade ago, and the expectation of encountering MTU-related problems is far lower than it was when a MSS based on a 1,500-octet MTU setting was used.

But there are still outstanding issues here, and a more reliable service can be staged using a slightly reduced local MTU and MSS setting. If we used a 1,480-octet MTU and corresponding TCP MSS values of 1,420 in IPv4 and 1,400 in IPv6, we could reasonably anticipate that the resultant TCP service would be adequately reliable.

As for the CVE mitigation advice to refuse a connection attempt when the remote-end MSS value is 500 or lower, I'd say that's good advice. It seems that the low MSS values are the result of some form of misconfiguration or error, and rather than attempting to mask over the error and persisting with an essentially broken TCP connection that is prone to generating a packet deluge, the best option is to just say "no" at the outset. If we all do that, then the misconfiguration will be quickly identified and fixed, rather than being silently masked over.

References and Further Reading

[1] CVE-2019-11477, National Vulnerability Database, Information Technology Laboratory, National Institute of Standards and Technology (NIST), June 2019.
`https://nvd.nist.gov/vuln/detail/CVE-2019-11477`

[2] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[3] J. Postel, "Internet Protocol," RFC 791, September 1981.

[4] J. Postel, "The TCP Maximum Segment Size and Related Topics," RFC 879, November 1983.

[5] Stephen E. Deering, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[6] David Borman, "TCP Options and Maximum Segment Size (MSS)," RFC 6691, July 2012.

[7] Geoff Huston, "A Tale of Two Protocols: IPv4, IPv6, MTUs and Fragmentation," *The ISP Column,* January 2009.
https://www.potaroo.net/ispcol/2009-01/mtu6.html

[8] Christopher A. Kent and Jeffrey C. Mogul, "Fragmentation Considered Harmful," Proceedings of Frontiers in Computer Communications Technology, ACM SIGCOMM '87, August 1987.

[9] Geoff Huston, "Fragmentation," *The Internet Protocol Journal,* Volume 19, No. 2, June 2016.

[10] Geoff Huston, "IPv6 and Packet Fragmentation," *The Internet Protocol Journal,* Volume 21, No. 1, April 2018.

GEOFF HUSTON, 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**

Looking Back on 50 Years of the Internet Era

by Vint Cerf, Internet Pioneer

It isn't possible in a short essay to really cover 50 years of the evolution of the Internet. Books have been written on the subject. We have just celebrated the October 29, 1969, milestone that linked an XDS Sigma-7 computer at the *University of California, Los Angeles* (UCLA) to a time-shared SDS 940 computer at *Stanford Research Institute* (SRI). At that time, there were only two nodes of the planned *Advanced Research Projects Agency Network* (ARPANET)[0], but two more at *University of California, Santa Barbara* (UCSB) and at *University of Utah*, *Salt Lake City* were added by the end of the year. What I think is most interesting about the ARPANET part of the Internet saga is the trailblazing that project did in heterogeneous computer networking. It was among the first networks to use *packet switching* as the communications mechanism. Some historians might reasonably argue that the US *Semi-Automated Ground Environment* (SAGE) system developed in the 1950s and in operation until the 1980s and the US *AUTODIN* message switching system also developed in the late 1950s and 1960s both represented then state-of-the-art, wide-area computer communications. These systems were closer in spirit to automated store-and-forward teletype/telegraph messaging services than to the subsequent packet switching of the ARPANET, the French *Cyclades/Cigale* experimental network, and the UK National Physical Laboratory local-area network.

The heterogeneous computers (called *hosts*) of the ARPANET were interconnected to each other through a subnet of identical *Interface Message Processors* (IMPs) that were, in turn, interconnected by dedicated 50 kbps telephone circuits.

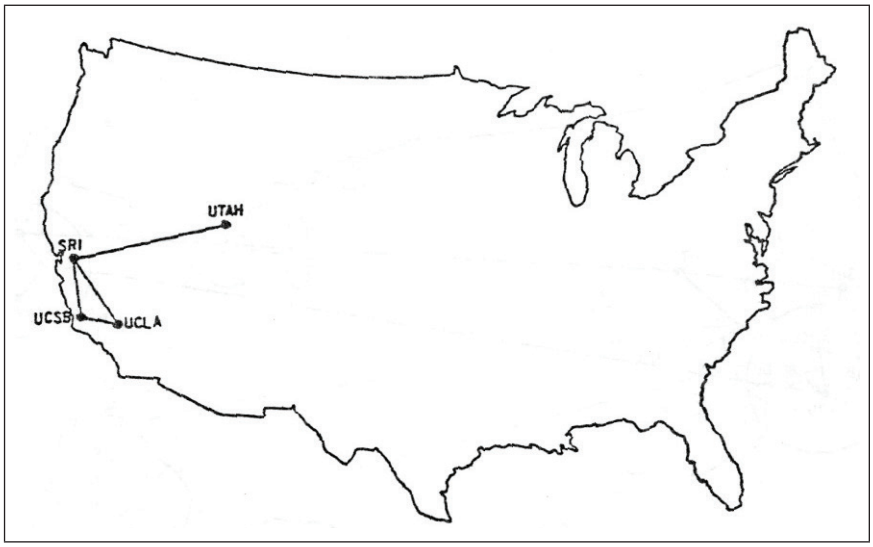


Figure 1: The ARPANET in December 1969[4]

The ARPANET project launched a documentation series called *Request for Comments* (RFCs)[1] that continues to this day to document the protocols of the Internet.

To coordinate the development of the host-to-host protocols and other applications, a *Network Working Group* (NWG) was established. This concept influenced the creation of an *International Network Working Group* (INWG) that became Working Group 6.1 of Technical Committee 6 of the *International Federation of Information Processing* (IFIP) and also influenced the creation of the *Internet Configuration Control Board* that morphed into the *Internet Architecture Board* (IAB), which gave rise to the *Internet Engineering and Research Taskforces* (IETF and IRTF), all of which are still active today.

A spirit of open sharing and cooperation permeated the participants and organizations that were involved in the ARPANET project and that also influenced the early developers of the Internet and the World Wide Web—which emerged in the early 1990s as the most popular application of the system. Even in today's competitive environment, we find the engineers of the Internet cooperating to deal with a plethora of malicious attacks against the infrastructure and applications of the Internet and to fashion new protocols to support a growing collection of applications.

As the protocol experiments and research unfolded on the ARPANET, the concept of protocol layering emerged and strongly influenced both the Internet design and the development of the *Open Systems Interconnection* (OSI) Model for computer networking. The layering concepts also influenced the basic Internet architecture in the form of encapsulation and decapsulation of Internet packets in the frames and packet payloads of lower-level protocols in the underlying networks of the Internet. Gateways received Internet packets in the payloads of lower layers, extracted them, decided where to route them, encapsulated them in the appropriate payloads of the next packet network, and sent them on their way.

Electronic messaging, which had been developed in the course of implementing time-shared computer systems, was extended as networked electronic mail in the early years of the ARPANET to work across the network among the cooperating hosts. A *File Transfer Protocol* (FTP)[2] and a remote-access *telecommunications network* (TELNET)[3] protocol were among the early applications of the ARPANET and were eventually translated into the Internet.

A look back to the early years of the ARPANET shows that we owe much to those pioneering researchers and engineers who blazed trails into terra incognita for the rest of us to follow and extend. Even today, there is still an enormous frontier of unexplored conceptual space waiting to be discovered. As we collectively struggle to deal with emergent challenges of misinformation, disinformation, denial-of-service attacks, and fragmentation of the Internet, I still remain hopeful that the utility of willing global collaboration will inform the Internet governance policies under consideration around the world and bend them towards positive and fruitful outcomes.

References and Further Reading

[0] "ARPANET," Wikipedia entry:
https://en.wikipedia.org/wiki/ARPANET

[1] "History," RFC Editor webpage:
https://www.rfc-editor.org/history/

[2] J. Postel and J. Reynolds, "File Transfer Protocol," RFC 959, October 1985.

[3] A. M. McKenzie, "Telnet Protocol Specifications," RFC 495, May 1973.

[4] Image courtesy of J. Noel Chiappa, MIT Advanced Network Architecture Group.

[5] Daniel Dern, "The ARPANET Is Twenty: What We Have Learned and The Fun We had," *ConneXions—The Interoperability Report*, Volume 3, No. 10, October 1989. Archive available from The Charles Babbage Institute at the University of Minnesota:
http://www.cbi.umn.edu/hostedpublications/Connexions/index.html

VINTON G. CERF is vice president and Chief Internet Evangelist for Google. He contributes to global policy development and continued spread of the Internet. Widely known as one of the "Fathers of the Internet," Cerf is the co-designer of the TCP/IP protocols and the architecture of the Internet. He has served in executive positions at MCI, the Corporation for National Research Initiatives and the Defense Advanced Research Projects Agency and on the faculty of Stanford University.

Vint Cerf served as chairman of the board of the *Internet Corporation for Assigned Names and Numbers* (ICANN) from 2000–2007 and has been a Visiting Scientist at the Jet Propulsion Laboratory since 1998. Cerf served as founding president of the *Internet Society* (ISOC) from 1992–1995. Cerf is a Foreign Member of the British Royal Society and Swedish Academy of Engineering, and Fellow of IEEE, ACM, and American Association for the Advancement of Science, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, the British Computer Society, the Worshipful Company of Information Technologists, the Worshipful Company of Stationers and a member of the National Academy of Engineering. He currently serves as Past President of the Association for Computing Machinery, Past Chairman of the *American Registry for Internet Numbers* (ARIN) and completed a term as Chairman of the Visiting Committee on Advanced Technology for the US National Institute of Standards and Technology. President Obama appointed him to the National Science Board in 2012.

Cerf is a recipient of numerous awards and commendations in connection with his work on the Internet, including the US *Presidential Medal of Freedom*, US *National Medal of Technology*, the *Queen Elizabeth Prize for Engineering*, the *Prince of Asturias Award*, the Tunisian *National Medal of Science*, the *Japan Prize*, the *Charles Stark Draper* award, the ACM *Turing Award*, Officer of the Legion d'Honneur and 29 honorary degrees. In December 1994, *People* magazine identified Cerf as one of that year's "25 Most Intriguing People." In 2012, he was inducted to the *Internet Hall of Fame*.
E-mail: **vint@google.com**

Book Review

Confessions of a Crypto Millionaire

Confessions of a Crypto Millionaire: My Unlikely Escape from Corporate America, by Dan Conway, Zealot Publishing, September 2019, ISBN-13: 978-1733171700.

You probably have read your fill of business books. Author tries to make it big, leverages tons of his money and time, hires the wrong people, fires them, then goes it alone before striking it rich and motoring off into the sunset in some expensive car. Dan Conway's *Confessions of a Crypto Millionaire* is not one of these books. Most business books offer just enough advice to fill a chapter, maybe two. Conway has a lot more to say about his obsession and investments in cryptocurrency, in particular *Ethereum*. Over a period of several years, he used his home mortgage equity loan and borrowed additional funds because he believed blockchain held the future model for decentralized corporations and the way that we will all work together. He ended up cashing out $14M ahead. It is his obsession that drives the book's narrative, along with the crazy up-and-down valuation of Ether, where you can gain and lose millions in a matter of minutes.

What isn't in this book is also notable: sordid tales of wretched excess of "tech-bros partying on yachts" or trashing expensive Vegas hotel suites. Conway is a father of three, and still married to their mother.

Conway's confessions make a refreshing tale about his fighting his demons, his addictions (alcohol and pills), his insecurities, and his almost always-on self-destructive alter-ego he calls his "Flip Side." This side rears its ugly head during client presentations where he fumbles and fails and during periods of self-doubt when he tries to reassure himself his huge bet on Ether isn't about to land him in the poor house.

"The book forced me to make sense of how my addictive personality played a part in my undoubtedly reckless crypto investments," he told me via an e-mail interview. He is part visionary, buying Ether at a time and at a level few people had the courage, vision, or just dumb luck to do. "It took everything admirable and loathsome about me to make the plunge into Ether. The loathsome part includes my addictive personality. While betting everything was an extreme risk, all risk requires insight, courage, and maybe a little recklessness." He hopes his story will get others to think about how they formulate their own risk taking.

Conway begins his story "working for the man," doing marketing and public relations for large corporations, one of whom he calls Acme. He wasn't a good fit as the organization man to be sure. And since his windfall with Ether, he is unlikely to return to corporate America "unless we suffer a financial catastrophe."

He still believes that the decentralized blockchain can disrupt the traditional corporate power structure and has a lot of merit as an organizing principle. One example he cites is the MakeDAO, where ordinary folks can originate loans and handle other financial transactions without any financial institutional limits. It could pay off; it could fall flat: that is the challenge of cryptocurrency.

One aspect of his book is dealing very honestly with two situations: first, with his addictions. "This undoubtedly played a part in my reckless crypto investments, and writing the book helped force me to make sense of it all."

Second, the book also describes how his financial windfall changed his family dynamics and the relationships with his circle of friends. Even though Conway lived in Silicon Valley, he was very firmly rooted in the middle class before he made it big with Ether. He writes:

"Crypto was suddenly like an overexposed celebrity, and everyone was rooting for it to fail," but then realizes, "one of the bittersweet feelings about making a bunch of money is that you can't bring your (less fortunate) friends with you." That takes some adjustment, both for him and his family. Still, don't be too sad: Now he takes long exotic vacations, buys his kids "name-brand clothes" instead of Sears knock-offs, and does car pool duty with a vengeance. "It's absolutely nice to have the car-ride conversations rather than pinning all parent/child bonding on the "how was your day?" question when everyone is exhausted." True that.

Conway is committed to Ethereum because of its disruptive ability to change the way companies operate, the way companies get Venture Capital funding (the parts about the ICO shysters alone are worth reading), and the way the early pioneers—which Conway counts as himself—had to try to separate the criminals from the legit businesses. This book is well worth reading, even if your own exposure to bitcoin and other cryptocoins is minimal.

—*David Strom*, **david@strom.com**

Ed.: This book review originally appeared in David Strom's *Web Informant*, available at: **https://blog.strom.com/wp/**

See also: William Stallings, "A Blockchain Tutorial," *The Internet Protocol Journal*, Volume 20, No. 3, November 2017.

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. For more information, contact us at **ipj@protocoljournal.org**

Letter to the Editor

Hi Geoff,

I have read the article "DNS Privacy and the IETF," in the latest issue of *Internet Protocol Journal* (Volume 22, No. 2, July 2019). Thank you for the excellent insights.

I am intrigued about the discussion of a world where apps would control the resolver. I am wondering how these apps would work in an IPv6-only world where the *Internet Service Provider* (ISP) does *Carrier Grade Network Address Translation* (CGNAT). If a DoH server responds with only an A record to say **`ipv4google.com`**, how does the handset make the connection? Or perhaps I misunderstand CGNAT—in reality the Internet-facing device/router has an RFC 1918 address and a global IPv6 address—IPv4 connectivity would use *Network Address and Port Translation* (NAPT) from the ISP public pool, whereas IPv6 connectivity can pass through without issue.

Thanks, — *Naveen Nathan*, **`naveen@lastninja.net`**

The author responds:

Great question. Some transition mechanisms "crossed the beams" and relied on a DNS resolver that had knowledge of the transition mechanism and deliberately lied in their responses in order to steer the end host's traffic to a protocol translator/encapsulator. Obviously if the application selected a DoH resolver that was not part of the local environment and was unaware of the need to provide NAT64 responses to these hosts, then the application would be unable to communicate.

It leads to the interesting outcome where the host (non-DoH) would look just fine and certain applications when going to certain remote services would fail. I have some sympathy for the help desk staff trying to identify and solve this problem.

Evidently some plans for DoH use involve application testing the existing configured DNS resolver for DoH capability and turning on DoH only in that case; that is, encapsulate in HTTP only the first DNS "hop" from the stub resolver to the recursive resolver. This solution would certainly avoid the NAT64 issue but would not really prevent the "my ISP is spying on my DNS transactions and possibly using this data in ways I am unaware of" scenario.

More generally, the more we adorn the network infrastructure and the more we add elements that create dependencies on other elements in novel ways, the more fragile the network becomes. The end point is a network that only barely functions and resists any modification—however slight—as the modification causes it to fail. It's an odd situation to get to when the original concepts behind the Internet were thoughts about creating a level of resiliency of the network as a service that exceeded the resiliency of any component of the network system.

Regards, — *Geoff Huston*, **`gih@apnic.net`**

Fragments

Postel Service Award Presented to Alain Aina

The Internet Society, a global nonprofit dedicated to ensuring the open development, evolution, and use of the Internet, recently presented the prestigious *Jonathan B. Postel Service Award* to Alain Aina, who serves as the chief technology officer of the *West and Central Africa Research and Education Network* (WACREN).

Aina has been building a Regional Research and Education Network to interconnect *National Research and Education Networks* (NRENs) in the region and connect them to the global Research and Education Network. He wants the world to see the work of Africa's premier researchers and carve out its spot in the academic world—in a way that would be impossible without the resources of this new network and community. He also contributes to *AfricaConnect2*, a project that supports the development of high-capacity networks for research and education across Africa, by building on existing networks in Eastern, Northern, and Southern Africa to connect to West and Central Africa's WACREN.

Photo by Minzayar Oo © IETF LLC 2019.

Aina fell into this work after graduating in the early 90s with a degree in electrical engineering and in the maintenance and analysis of computer systems. He was hired to be a technical seller for a company in the Togolese Republic, which had a branch in Benin, where he is from. The owner of the company had recently returned home from the United States and was anxious about computing and internetworking. He noticed Aina's talent and added him to the technical team, where he ended up building the first *Bulletin Board System* (BBS) in the area.

"People used the modem to dial in, then people on the same server could talk to each other," he said. "Then we decided to put in the first e-mail gateway, connecting to someone in Accra and later in Montreal twice a day to drop mail and download mail. But the cost was so high, it was not sustainable. The delegation of the country-code TLD in 1996 changed the paradigm for the e-mail service and we were proud to demonstrate the first local web server and intranet."

By the mid-90s there wasn't a lot of support for people working on Internet access and connection, but there was ever-growing interest and demand. This meant that Aina and his colleagues often worked around the clock to set up networks and services in communities, then trained the local population on how to use what they had made.

"The Internet became so popular that the demand was suddenly so high, and it was putting pressure on us," he said.

It was about this time that Aina started collaborating with the *Network Startup Resource Center* (NSRC), where he now serves as a part-time network engineer and trainer. He later launched the first full IP services in the Togolese Republic and then in other countries in West Africa.

"At that time, most of the world did not believe that Africa could have the Internet and play a role. When you'd go to places, you'd have to train people," Aina said. "Training materials were rare. We were lucky to have some books and some knowledgeable friends far away. The people you trained only knew you, so if something broke they called you to fix it."

Aina helped build large parts of the Internet ecosystem throughout Africa, setting up networks, contributing to the creation of the regional Internet registry and the network operator group, and building ccTLD registries. He also started a consulting firm and became active in the private sector.

He eventually started attending *Internet Society* (ISOC) network technology workshops and getting involved with the organization in other ways. From 2011 to 2014, he served as a trustee for the organization. Active in the Internet community, he is also involved with ICANN, the *African Network Information Centre* (AFRINIC), the *African Network Operators Group* (AFNOG), and other organizations. He helped found AFNOG, where he's been an instructor since 2000, and he is one of the founders of AFRINIC, where he's served in several roles, including acting chief technology officer, acting chief executive officer, and director of research and innovation. Aina is a key technical resource for the DNS community, including *Africa Top Level Domains Organization* (AFTLD). A big part of his life has been Internet related, but he feels there is still so much more to do for Africa.

Mr. Aina was selected by an international award committee comprised of former Postel award winners. The committee placed particular emphasis on candidates who have supported and enabled others in addition to their own contributions. The award was presented to Mr. Aina in recognition of his leadership in pioneering the Internet in Africa and building technical communities that helped connect countless others across the continent and beyond. Aina helped build large parts of the Internet ecosystem throughout Africa, setting up networks, contributing to the creation of the regional Internet registry and the network operator group, and building ccTLD registries.

"This award encourages me to continue the work, to grow and help others spread the Internet continent-wide, and to help break down barriers for the engineers and scientists in Africa," Aina said. "I feel happy and honored to be recognized for this work."

The Postel Award was established by the Internet Society to honor individuals or organizations that, like Jon Postel, have made outstanding contributions to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. Andrew Sullivan, President and CEO of the Internet Society presented the award, including a US$20,000 honorarium and a crystal engraved globe, during the 106th meeting of the *Internet Engineering Task Force* (IETF) held in Singapore, 16–22 November 2019.

ICANN Calls for Full DNSSEC Deployment

The *Internet Corporation for Assigned Names and Numbers* (ICANN) believes that there is an ongoing and significant risk to key parts of the *Domain Name System* (DNS) infrastructure. In the context of increasing reports of malicious activity targeting the DNS infrastructure, ICANN is calling for full deployment of the *Domain Name System Security Extensions* (DNSSEC) across all unsecured domain names. The organization also reaffirms its commitment to engage in collaborative efforts to ensure the security, stability and resiliency of the Internet's global identifier systems.

As one of many entities engaged in the decentralized management of the Internet, ICANN is specifically responsible for coordinating the top-most level of the DNS to ensure its stable and secure operation and universal resolvability.

On 15 February 2019, in response to reports of attacks against key parts of the DNS infrastructure, ICANN offered a checklist[1] of recommended security precautions for members of the domain name industry, registries, registrars, resellers, and related others, to proactively take to protect their systems, their customers' systems and information reachable via the DNS.

Public reports[2] indicate that there is a pattern of multifaceted attacks utilizing different methodologies. Some of the attacks target the DNS, in which unauthorized changes to the delegation structure of domain names are made, replacing the addresses of intended servers with addresses of machines controlled by the attackers. This particular type of attack, which targets the DNS, only works when DNSSEC is not in use. DNSSEC is a technology developed to protect against such changes by digitally "signing" data to assure its validity. Although DNSSEC cannot solve all forms of attack against the DNS, when it is used, unauthorized modification to DNS information can be detected, and users are blocked from being misdirected.

ICANN has long recognized the importance of DNSSEC and is calling for full deployment of the technology across all domains. Although this will not solve the security problems of the Internet, it aims to assure that Internet users reach their desired online destination by helping to prevent so-called *Man in the Middle* attacks where a user is unknowingly re-directed to a potentially malicious site. DNSSEC complements other technologies, such as *Transport Layer Security* (TLS) (most typically used in HTTPS) that protect the end user/domain communication.

As the coordinator of the top-most level of the DNS, ICANN is in the position to help mitigate and detect DNS-related risks, and to facilitate key discussions together with its partners. The organization believes that all members of the domain name system ecosystem must work together to produce better tools and policies to secure the DNS and other critical operations of the Internet.

[1] "Alert Regarding Published Reports of Attacks on the Domain Name System,"
https://www.icann.org/news-announcement-2019-02-15-en

[2] "A Deep Dive on the Recent Widespread DNS Hijacking Attacks," Krebs on Security, February 19, 2019.
https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/

The RIPE NCC has run out of IPv4 Addresses

From the RIPE-NCC Website: "Today, at 15:35 (UTC+1) on 25 November 2019, we made our final `/22` IPv4 allocation from the last remaining addresses in our available pool. We have now run out of IPv4 addresses. Our announcement will not come as a surprise for network operators—IPv4 run-out has long been anticipated and planned for by the RIPE community. In fact, it is due to the community's responsible stewardship of these resources that we have been able to provide many thousands of new networks in our service region with `/22` allocations after we reached our last `/8` in 2012.

Even though we have run out, we will continue to recover IPv4 addresses in the future. These will come from organisations that have gone out of business or are closed, or from networks that return addresses they no longer need. These addresses will be allocated to our members (*Local Internet Registries* [LIRs]) according to their position on a new waiting list that is now active.

While we therefore expect to be allocating IPv4 for some time, these small amounts will not come close to the many millions of addresses that networks in our region need today. Only LIRs that have never received an IPv4 allocation from the RIPE NCC (of any size) may request addresses from the waiting list, and they are only eligible to receive a single `/24` allocation. LIRs that have submitted an IPv4 request can see their position on the waiting list in the LIR Portal. A graph is available at `https://www.ripe.net/` that shows the number of requests on the waiting list and the number of days that the LIR at the front of the queue has been waiting.

This event is another step on the path towards global exhaustion of the remaining IPv4 addressing space. In recent years, we have seen the emergence of an IPv4 transfer market and greater use of *Carrier Grade Network Address Translation* (CGNAT) in our region. There are costs and trade-offs with both approaches and neither one solves the underlying problem, which is that there are not enough IPv4 addresses for everyone.

Without wide-scale IPv6 deployment, we risk heading into a future where the growth of our Internet is unnecessarily limited—not by a lack of skilled network engineers, technical equipment or investment—but by a shortage of unique network identifiers. There is still a long way to go, and we call on all stakeholders to play their role in supporting the IPv6 roll-out.

At the RIPE NCC, we are here to support our membership and the wider RIPE community in this work. Aside from allocating the IPv6 resources that will be required, we will continue to provide advice, training, measurements and tools to help network operators as they put their deployment plans into action. We are optimistic and excited to see what the next chapter will bring. So let's get to work—and together, let's shape the future of the Internet."

Check your Subscription Details!

If you have a print subscription to this journal, you will find an expiration date printed on the back cover. For the last couple of years, we have "auto-renewed" your subscription, but now we ask you to log in to our subscription system and perform this simple task yourself. The subscription portal is here: **https://www.ipjsubscription.org/** This process will ensure that we have your current contact information as well as delivery preference (print edition or download). For any questions, contact us by e-mail at: **ipj@protocoljournal.org**

Our Privacy Policy

The *General Data Protection Regulation* (GDPR) is a regulation for data protection and privacy for all individual citizens of the *European Union* (EU) and the *European Economic Area* (EEA). Its implementation in May 2018 led many organizations worldwide to post or update privacy statements regarding how they handle information collected in the course of business. Such statements tend to be long and include carefully crafted legal language. We realize that we may need to provide similar language on our website and in the printed edition, but until such a statement has been developed here is an explanation of how we use any information you have supplied relating to your subscription:

- The mailing list for *The Internet Protocol Journal* (IPJ) is entirely "opt in." We never have and never will use mailing lists from other organizations for any purpose.
- You may unsubscribe at any time using our online subscription system or by contacting us via e-mail. We will honor any request to remove your name and contact information from our database.
- We will use your contact information *only* to communicate with you about your subscription; for example, to inform you that a new issue is available, that your subscription needs to be renewed, or that your printed copy has been returned to us as undeliverable by the postal authorities.
- We will *never* use your contact information for any other purpose or provide the subscription list to any third party other than for the purpose of distributing IPJ by post or by electronic means.
- If you make a donation in support of the journal, your name will be listed on our website and in print unless you tell us otherwise.

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**

Fabrizio Accatino
Michael Achola
Martin Adkins
Christopher Affleck
Scott Aitken
Jacobus Akkerhuis
Antonio Cuñat Alario
Nicola Altan
Matteo D'Ambrosio
Jens Andersson
Danish Ansari
Tim Armstrong
Richard Artes
Michael Aschwanden
David Atkins
Jac Backus
Jaime Badua
Eric Baker
Santosh Balagopalan
David Belson
Hidde Beumer
Pier Paolo Biagi
John Bigrow
Orvar Ari Bjarnason
Axel Boeger
Keith Bogart
Mirko Bonadei
Roberto Bonalumi
Julie Bottorff Photography
Gerry Boudreaux
L de Braal
Kevin Breit
Thomas Bridge
Ilia Bromberg
Václav Brožík
Christophe Brun
Gareth Bryan
Stefan Buckmann
Caner Budakoglu
Darrell Budic
Scott Burleigh
Jon Harald Bøvre
Olivier Cahagne
Antoine Camerlo
Tracy Camp
Ignacio Soto Campos
Fabio Caneparo
Roberto Canonico
David Cardwell
John Cavanaugh
Lj Cemeras
Dave Chapman
Stefanos Charchalakis
Greg Chisholm
David Chosrova
Marcin Cieslak
Brad Clark
Narelle Clark
Steve Corbató
Brian Courtney
Dave Crocker
Kevin Croes
John Curran
André Danthine
Morgan Davis
Jeff Day
Julien Dhallenne
Freek Dijkstra
Geert Van Dijk
David Dillow
Richard Dodsworth
Ernesto Doelling
Michael Dolan
Eugene Doroniuk
Karlheinz Dölger
Joshua Dreier
Lutz Drink
Andrew Dul
Joan Marc Riera Duocastella
Holger Durer
Mark Eanes
Peter Robert Egli
George Ehlers
Peter Eisses
Torbjörn Eklöv
Y Ertur
ERNW GmbH
ESdatCo
Steve Esquivel
Jay Etchings
Mikhail Evstiounin
Paul Ferguson
Ricardo Ferreira
Kent Fichtner
The Flirble Organisation
Gary Ford
Jean-Pierre Forcioli
Susan Forney
Christopher Forsyth
Andrew Fox
Craig Fox
Fausto Franceschini
Tomislav Futivic
Edward Gallagher
Andrew Gallo
Chris Gamboni
Xosé Bravo Garcia
Osvaldo Gazzaniga
Kevin Gee
Greg Giessow
John Gilbert
Serge Van Ginderachter
Greg Goddard
Tiago Goncalves
Octavio Alfageme Gorostiaga
Barry Greene
Richard Gregor
Martijn Groenleer
Geert Jan de Groot
Christopher Guemez
Gulf Coast Shots
Sheryll de Guzman
Jason Hall
James Hamilton
Stephen Hanna
Martin Hannigan
John Hardin
David Harper
Edward Hauser
David Hauweele
Marilyn Hay
Headcrafts SRLS
Hidde van der Heide
Johan Helsingius
Robert Hinden
Asbjorn Hojmark
Damien Holloway
Alain Van Hoof
Edward Hotard
Bill Huber
Hagen Hultzsch
Kevin Iddles
Mika Ilvesmaki
Karsten Iwen
David Jaffe
Ashford Jaggernauth
Martijn Jansen
Jozef Janitor
John Jarvis
Dennis Jennings
Edward Jennings
Aart Jochem
Brian Johnson
Curtis Johnson
Richard Johnson
Jim Johnston
Jonatan Jonasson
Daniel Jones
Gary Jones
Jerry Jones
Anders Marius Jørgensen
Amar Joshi
Merike Kaeo
Andrew Kaiser
Christos Karayiannis
David Kekar
Jithin Kesavan
Jubal Kessler
Shan Ali Khan
Nabeel Khatri
Dae Young Kim
Russell Kirk
Anthony Klopp
Henry Kluge
Michael Kluk
Andrew Koch
Ia Kochiashvili
Carsten Koempe
Richard Koene
Alexader Kogan
Antonin Kral
Mathias Körber
Robert Krejčí
John Kristoff
Terje Krogdahl
Bobby Krupczak
Murray Kucherawy
Dirk Kurfuerst
Warren Kumari
Darrell Lack
Yan Landriault
Markus Langenmair
Fred Langham
Andrew Lamb
Richard Lamb
Sig Lange
Tracy LaQuey Parker
Rick van Leeuwen
Simon Leinen
Robert Lewis
Martin Lillepuu
Roger Lindholm
Sergio Loreti
Eric Louie
Guillermo a Loyola
Hannes Lubich
Dan Lynch
Miroslav Madić
Alexis Madriz
Carl Malamud
Jonathan Maldonado
Michael Malik
Yogesh Mangar
Bill Manning
Harold March
Vincent Marchand
Gabriel Marroquin
David Martin
Jim Martin
Ruben Tripiana Martin
Timothy Martin
Juan Jose Marin Martinez
Carles Mateu
Ioan Maxim
David Mazel
Miles McCredie
Brian McCullough
Joe McEachern
Alexander McKenzie
Jay McMaster
Mark Mc Nicholas
Carsten Melberg
Kevin Menezes
Bart Jan Menkveld
William Mills

David Millsom
Desiree Miloshevic
Joost van der Minnen
Thomas Mino
Rob Minshall
Wijnand Modderman
Mohammad Moghaddas
Charles Monson
Andrea Montefusco
Fernando Montenegro
Joel Moore
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
Tim O'Brien
Mike O'Connor
Mike O'Dell
Jim Oplotnik
Carlos Astor Araujo Palmeira
Alexis Panagopoulos
Gaurav Panwar
Manuel Uruena Pascual
Ricardo Patara
Dipesh Patel
Alex Parkinson
Craig Partridge
Dan Paynter
Leif Eric Pedersen
Rui Sao Pedro
Juan Pena
Chris Perkins
Michael Petry
Alexander Peuchert
David Phelan
Derrell Piper
Rob Pirnie
Marc Vives Piza
Jorge Ivan Pincay Ponce
Victoria Poncini
Blahoslav Popela
Eduard Llull Pou
Tim Pozar
David Raistrick
Priyan R Rajeevan
Balaji Rajendran
Paul Rathbone
William Rawlings
Bill Reid
Petr Rejhon
Robert Remenyi
Rodrigo Ribeiro
Glenn Ricart
Justin Richards
Mark Risinger
Ron Rockrohr
Carlos Rodrigues
Magnus Romedahl
Lex Van Roon
Alessandra Rosi
William Ross
Boudhayan Roychowdhury
Carlos Rubio
Timo Ruiter
RustedMusic
Babak Saberi
George Sadowsky
Scott Sandefur
Sachin Sapkal
Arturas Satkovskis
PS Saunders
Richard Savoy
John Sayer
Phil Scarr
Elizabeth Scheid
Jeroen Van Ingen Schenau
Carsten Scherb
Ernest Schirmer
Dan Schrenk
Richard Schultz
Roger Schwartz
SeenThere
Scott Seifel
Yury Shefer
Yaron Sheffer
Doron Shikmoni
Tj Shumway
Jeffrey Sicuranza
Thorsten Sideboard
Greipur Sigurdsson
Andrew Simmons
Pradeep Singh
Henry Sinnreich
Geoff Sisson
Helge Skrivervik
Darren Sleeth
Richard Smit
Bob Smith
Courtney Smith
Mark Smith
Job Snijders
Ronald Solano
Asit Som
Ignacio Soto Campos
Evandro Sousa
Peter Spekreijse
Thayumanavan Sridhar
Paul Stancik
Ralf Stempfer
Matthew Stenberg
Adrian Stevens
Clinton Stevens
John Streck
Martin Streule
Viktor Sudakov
Edward-W. Suor
Vincent Surillo
T2Group
Roman Tarasov
David Theese
Douglas Thompson
Lorin J Thompson
Joseph Toste
Rey Tucker
Sandro Tumini
Angelo Turetta
Phil Tweedie
Steve Ulrich
Unitek Engineering AG
John Urbanek
Martin Urwaleck
Betsy Vanderpool
Surendran Vangadasalam
Ramnath Vasudha
Philip Venables
Buddy Venne
Alejandro Vennera
Luca Ventura
Tom Vest
Dario Vitali
Michael L Wahrman
Laurence Walker
Randy Watts
Andrew Webster
Tim Weil
Jd Wegner
Westmoreland Engineering Inc.
Rick Wesson
Peter Whimp
Russ White
Jurrien Wijlhuizen
Derick Winkworth
Pindar Wong
Janko Zavernik
Muhammad Ziad Ziayuddin
Romeo Zwart
Bernd Zeimetz
廖 明沂.

Call for Papers

The *Internet Protocol Journal* (IPJ) is a quarterly technical publication containing tutorial articles ("What is...?") as well as implementation/operation articles ("How to..."). The journal provides articles about all aspects of Internet technology. IPJ is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. In addition to feature-length articles, IPJ contains technical updates, book reviews, announcements, opinion columns, and letters to the Editor. Topics include but are not limited to:

- Access and infrastructure technologies such as: Wi-Fi, Gigabit Ethernet, SONET, xDSL, cable, fiber optics, satellite, and mobile wireless.
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance.
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping.
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, cloud computing, and quality of service.
- Application and end-user issues such as: E-mail, Web authoring, server technologies and systems, electronic commerce, and application management.
- Legal, policy, regulatory and governance topics such as: copyright, content control, content liability, settlement charges, resource allocation, and trademark disputes in the context of internetworking.

IPJ will pay a stipend of US$1000 for published, feature-length articles. For further information regarding article submissions, please contact Ole J. Jacobsen, Editor and Publisher. Ole can be reached at **ole@protocoljournal.org** or **olejacobsen@me.com**

Supporters and Sponsors

Supporters

Internet Society

cisco

Diamond Sponsors

Google

ICANN

Ruby Sponsors

Your logo here!

Sapphire Sponsors

Your logo here!

Emerald Sponsors

Afilias

Akamai

APNIC

APRICOT Asia Pacific Regional Internet Conference on Operational Technologies

COMCAST

DE-CIX

EQUINIX

JPRS Japan Registry Services

Juniper Networks

lacnic

LinkedIn

linx

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

amsix amsterdam internet exchange

DNS-OARC

IWL

ISC Internet Systems Consortium

Limelight Networks

PKNIC

qa cafe

SIDN

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

The Internet Protocol Journal
NMS
535 Brennan Street
San Jose, CA 95131

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