The Internet Protocol *Journal*

September 2020 Volume 23, Number 2

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

In June 2013 we published an article entitled "Optimizing Link-State Protocols for Data Center Networks." In that article, Alvaro Retana and Russ White wrote: "With the advent of cloud computing, the pendulum has swung from focusing on wide-area or global network design toward a focus on *Data Center* network design. Many of the lessons we have learned in the global design space will be relearned in the data center space before the pendulum returns and wide-area design comes back to the fore." In this issue, Russ White and Melchior Aelmans examine the use of link-state alternatives to the *Border Gateway Protocol* (BGP) in data center designs. One such alternative, *Routing in Fat Trees* (RIFT), will be explored further in an upcoming article in this journal, so please make sure your subscription details are up-to-date.

The depletion of the IPv4 address space and transition to IPv6 has been covered in numerous articles in IPJ over more than two decades. It was initially believed that "everyone" would implement IPv6 by the time the *Regional Internet Registries* (RIRs) ran out of addresses, but such predictions have proven to be too optimistic for a variety of reasons. The demand for public IPv4 address space has led to an "aftermarket," whereby blocks of addresses can be purchased (or leased) through the use of address brokers. We asked David Strom to explore this market in more detail, and he approached this assignment by deciding to sell his own Class C address block.

We are excited to bring you another book review, this time on the topic of information security. Please send us your suggestions for networking-related books that we should have reviewed.

Publication of *The Internet Protocol Journal* is made possible by the generous support of numerous individuals and organizations. Please consider making a donation or getting your company to sign up for a sponsorship.

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`

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

Recent Developments in Link State on Data-Center Fabrics

by Russ White and Melchior Aelmans, Juniper Networks

Since the initial publication of the drafts resulting in RFC 7938[0], the *Border Gateway Protocol* (BGP)[10] has been the default choice for *Data-Center* (DC) fabrics, assumed by most controllers, intent-based systems, training courses in DC fabrics, and implementers. Recent activity in the *Internet Engineering Task Force* (IETF) and implementers suggests using link-state protocols in DC fabrics. This article explores why this move towards link-state protocols on DC fabrics is taking place, and then considers three specific avenues to link state on DC fabrics: *Distributed* (or localized) *Optimized Flooding* in *Intermediate System-to-Intermediate System Protocol* (IS-IS)[1], centralized calculation of optimal flooding trees[2, 3], and *Routing in Fat Trees* (RIFT)[4].

The arguments presented in this article are legitimate reasons not to use BGP for the DC fabric underlay and show that options other than BGP are available. Readers might (incorrectly) conclude the authors believe BGP should never be used as the routing protocol for a DC fabric overlay—but that is not true. To make a case for link-state protocols in DC fabric underlays, an extensive examination of the positive and negative aspects of BGP—and the other available protocols—is essential. Ultimately, it is up to individual operators to decide which protocol is "the best" for their application, a decision based on business and operational—as well as technical—reasons.

Defining Terms

Defining terms is often considered pedantic, and therefore often overlooked. But as the networking world spreads to wholly virtual environments, definitions quickly become blurry and local. In this article, two distinct terms that might be used differently in other contexts are used. The first is the *underlay*. For this article, the underlay is the physical infrastructure, control plane, and telemetry; it provides basic connectivity, including IPv4, IPv6, and *Multiprotocol Label Switching* (MPLS), edge-to-edge in the fabric. The *overlay*, on the other hand, provides virtual topologies which tunnel traffic edge-to-edge through fabric-side interfaces and devices. In other words, the overlay consists of tunnels with head- and tail-ends on *Top of Rack* (ToR, or *leaf*) switches or servers attached to the fabric and the control planes that provide reachability through those tunnels.

In other words, underlay control planes do not carry overlay reachability, overlay control planes do not carry underlay reachability, and underlay devices (other than where they terminate an overlay tunnel) do not switch based on overlay destinations. If this explanation sounds vaguely like the *Exterior Gateway Protocol* (EGP) versus *Interior Gateway Protocol* (IGP) split in a traditional transit provider network, that is because it is—just like the EGP/IGP split in a conventional transit network.

The underlay/overlay distinction might be confusing in some discussions because people who work entirely within cloud services may well consider the set of tunnels built between virtual machines or containers the overlay, and everything under these tunnels the underlay—even if the network has two layers of tunnels. There is no set of standard terms for the situation where a bottom layer provides connectivity based on the physical fabric topology, a collection of virtual networks on top of that used to create logical topologies, and another set of virtual networks within those logical topologies formed by the applications running over the network. Overlay tends to end up being used for both the "middle layer" and the "upper layer," hence the importance of defining the terms as they are used here.

Two other terms of importance here are *autonomic* and *automatable*. Confusion around these terms arises from the use of *Zero Touch Provisioning* (ZTP), used to mean the configuration of a device that does not need manual configuration to deploy. While both automated and autonomic networks are ZTP, there is still a difference between the two concepts. The closer a protocol comes to not needing any configuration at all, whether that configuration is automated or not, the closer the protocol is to being autonomic. While autonomic and automatable protocols appear similar, there are differences. The automation system must still be managed and maintained, there are still interfaces to integrate and manage, etc. Autonomic control planes may (or may not) be more complex, at least under the surface, than automatable ones; regardless, they are not the same thing.

It is rare for a protocol to be fully automated or fully autonomic; these two are a continuum rather than a binary space. For instance, BGP is not autonomic by design but can be modified to allow BGP speakers to discover one another and form a peering relationship automatically. In other cases, it might be possible to derive information, such as the IS-IS system ID, automatically, but doing so might make troubleshooting and maintenance more difficult—so automatic assignment might be possible, but not always desirable. Individual operators may have different optimal positions along the automatable-to-autonomic continuum.

Given autonomic to automatable is a spectrum of options, why would you choose to move towards autonomic operation? After an automation system is put in place to support the network, it may not seem to make much difference.

In a sense, moving from automatable to autonomic is simply shifting complexity from one place in the network to another. Moving configuration from the automation system to the protocol moves complexity from the automation system to the protocol as well. The one vital difference is each piece of complexity moved from the automation system to the protocol is one less interface to manage, one less piece of state the automation system must build and keep track of, etc.

Complex automation systems can be difficult to create and manage. Even in fully automated networks, research shows a major portion of network failures are caused by configuration failures.[5] Deducing the amount of state the automation system, and the humans supporting the automation system, is managing can be justified by reducing the number of places mistakes can happen. The more the protocol can figure out on its own, the less you must figure out how to configure.

This article assumes *spine-and-leaf* (or leaf-and-spine!) fabrics. A *Clos*[14] is a three-stage fabric, while a five-stage fabric wired in the most common way is called a *butterfly*. Butterfly fabrics are illustrated in two ways; "as-wired" with the leaves on both the top and bottom of the diagram and "folded" with the leaves arrayed at the bottom of the diagram.[6]

The number of stages in a spine-and-leaf fabric denotes the maximum number of switches a packet is forwarded through when crossing from edge to edge. A spine-and-leaf fabric may have multiple spines; a three-stage fabric has one spine, while a five-stage fabric has three spine stages. The inner or top-most tier (depending on how the fabric is drawn) is considered the *Top of Fabric* (ToF) or the fabric layer. In a five-stage fabric, the "middle tier" is called either the *Top of Pod* (ToP) or the *spine*. The level or tier denotes the distance from the edge, with the ToR or leaf nodes being considered T0, the "middle" stage T1, and the fabric or ToF stage T2.

BGP in the Underlay

For the last 10 to 15 years, BGP has been the "underlay protocol of choice" for DC fabrics. While the reasons for using BGP in the underlay have been outlined in several places through the years, including RFC 7938[0], it is useful to recap and explore some of these reasons as background.

First, BGP is widely implemented; virtually every routing vendor and every open-source routing stack such as *Free Range Routing* (FRRouting) has a fairly complete and well-tested BGP implementation. You can be confident that no matter whose hardware and software you choose, BGP will be supported—and the implementation is likely to be mature, interoperable with other implementations, and running in production in a lot of networks.

Second, BGP was—at least at one time—conceived of as one of the most straightforward routing protocols to understand and implement well. The logic of path-vector is reasonably easy to implement correctly, and the underlying transport mechanism, the *Transmission Control Protocol* (TCP), is built into every operating system already.

Third, BGP is widely deployed, and hence well understood by operators. Operators consider it easier to hire someone who knows BGP than one who knows any other protocol, and it is easy to find tooling for operating BGP in the open-source community.

There is a bit of irony in this point as 10 years ago it was almost impossible to find engineers with solid BGP experience; the advent of BGP on large-scale data-center fabrics has become something of a "self-fulfilling prophecy" in this regard.

Fourth, where scale is of concern, the perception is BGP outshines every other protocol. After all, "BGP runs the global Internet," and you cannot ask for a better proof point of scalability than that. The initial implementations of BGP on large-scale DC fabrics originally tried various IGPs, and found they could not scale to the size required.

Fifth, BGP has extensive prefix-filtering, route-tagging, and traffic-engineering capabilities. No other protocol, other than perhaps *Enhanced Interior Gateway Routing Protocol* (EIGRP) (!), can match the ability of BGP to control route flow.

Sixth, you can use BGP for both the underlay and the overlay in a single network. In theory, this possibility makes the configuration simpler. The normal configuration when using BGP for both is to configure the underlay using *External BGP* (eBGP) peering and the overlay as *Interior BGP* (iBGP) peering.

With all these advantages, why would you decide to move away from using BGP in both the underlay and overlay?

Challenging BGP in the Underlay

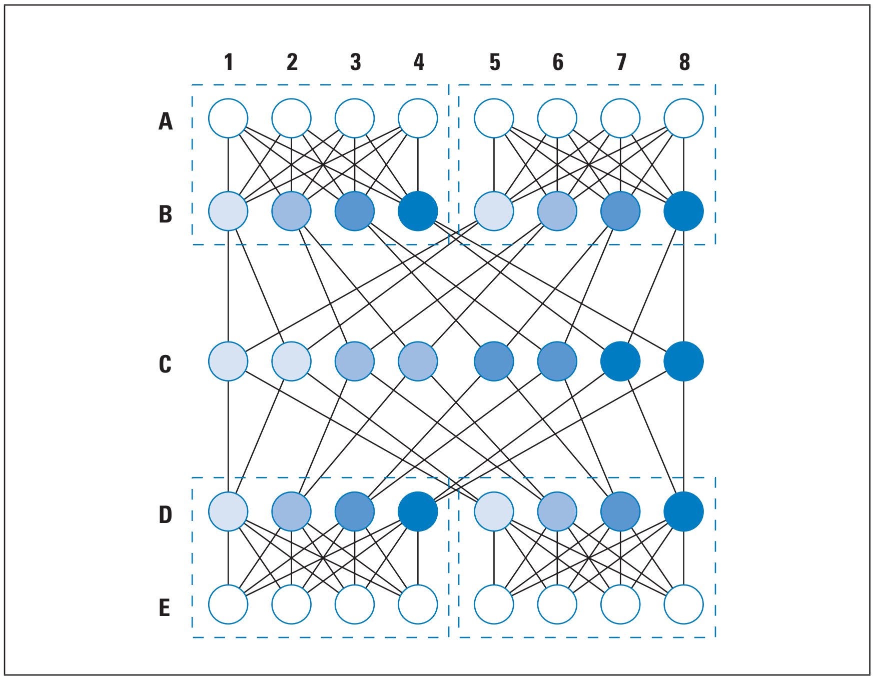
There are, however, counterpoints to many of the advantages of using BGP as the underlay protocol listed previously. Beginning with the second one—BGP is one of the simplest routing stacks to implement. With the advent of multiple address families, the *Resource Public Key Infrastructure* (RPKI), *Ethernet VPN* (EVPN), *Virtual Private LAN Service* (VPLS), MPLS traffic engineering, *BGP Link-State* (BGP-LS), and the many other features that have been "piled into" BGP across the last 20 years, BGP implementations have exploded in complexity. BGP may be the most complex protocol to implement among all the routed control planes today.

Using BGP as a singular DC fabric protocol, both overlay and underlay, is one factor causing the increasing complexity of BGP implementations. The ability to peer on unnumbered interfaces, the ability to accept any peer with any *Autonomous System* (AS) number, the ability to accept routes without any filters implemented, and many other changes must be made to make BGP work correctly in a DC fabric. It is easy enough to create a single knob that turns on a group of features at once. It is not so easy to hide the increased complexity—and the higher chance of a defect in the code or a misconfiguration of some kind—resulting from these kinds of changes. BGP is strongly automatable, but it will take massive work to make it autonomic. Is pushing that work into code used at critical points throughout the Internet a good idea?

At some point, the routing community needs to decide if it is wise to make one protocol the "protocol to end all protocols." Is a single solution the right answer for all problems? Or is it better to move back towards developing multiple parallel protocols to support different purposes? This criticism may not apply to operators building their private implementation of BGP for use on their DC fabrics—but these kinds of implementations are uncommon.

A second related issue is the amount of specialized configuration required to allow BGP to converge quickly on the kinds of dense topologies used for DC fabrics. Figure 1 illustrates the design.

Figure 1: A Small Butterfly Fabric



Note that in this diagram, A and E are ToR switches or leaf nodes, B and D are spines, and C is either the superspine or fabric. Dashed boxes around a set of devices indicate the pods. How BGP converges depends on the kind of topology change. In the case of a single router or link failure, BGP can converge almost as quickly as an IGP, given the failure timers are tuned correctly, BFD and other underlying mechanisms are in use, etc. The case of a withdrawal from the edge of the network, however, is much different.

In the case of a withdrawal, BGP converges by hunting across available paths, starting from the shortest and ending in the longest. This hunt does not happen because of the way BGP is designed, but rather because of the timing of processing and forwarding updates. To prevent loops, a BGP speaker must process an update locally, modifying the routing table before it can forward the update to its peers. Longer paths just take longer than shorter ones for withdrawals to traverse. This withdrawal behavior can be a problem in at least two situations: when a workload is moved from one location on the fabric to another, and when an anycast address representing a service instance is removed from the fabric.

In these cases, the slow convergence time of BGP can impact applications running on the fabric.

Controlling the impact of the hunt is fairly easy. The key is to reduce the length of the paths through which BGP must "search." The easiest way to do it is to block the reflection of updates and withdrawals through the network. For instance, E1 in Figure 1 should not reflect any withdrawals or updates to any of its peers in row D, and D1 should not reflect any updates or withdrawals to any of its peers in row C. There are many ways to accomplish these stipulations, but a common method is to create filters on the routers at rows A and E, the leaf nodes or ToR switches, so only BGP updates with an empty AS path (^$) are permitted, and to place all the routers at the spine routers (such as B and D) within a single pod in the same AS.

With these changes, BGP is essentially converted into "Fancy *Routing Information Protocol* (RIP)," and you can reduce the time required to withdraw a route (or move it from one place to another in the fabric) to about 1 minute in large-scale fabrics. It is possible to modify BGP to converge more quickly, but doing that returns the discussion to the first argument discussed previously—is creating a single protocol to solve all problems really the right answer? When is the complexity of the BGP code "complex enough" to start considering other options?

Let's examine two other considerations before moving on to examining link-state protocols in DC fabric underlays. One of the advantages listed for BGP is that it has many different policy options, such as route filtering and tagging. If the underlay is really designed to provide undifferentiated IP connectivity, these policies do not seem like much of a real advantage. Policy, such as route tagging and filtering, should be moved to the overlay—which is most likely going to be BGP anyway.

A final point is that transit providers separate infrastructure and customer routes to split these two kinds of information into different failure domains. One misunderstanding about failure domains is they must be "absolute" and "complete," where the two failure domains are completely decoupled at every point, if they are effective. They are not, however, always effective because it is likely impossible to build networks out of completely decoupled failure domains. Instead, it is a matter of tradeoffs. How much gain is there in separating these two kinds of information in this way, versus how difficult is it to separate these two kinds of information, and how much deoptimization is likely to occur?

In a DC fabric, separating the infrastructure routes of the underlay from the "customer" routes in the overlay is a legitimate way to form two different failure domains. These two failure domains might be somewhat tightly coupled, but they are still two different failure domains.

Separating the routes this way also creates multiple administrative domains, leaving open the possibility of allowing "customers," or workload processes, to control some aspects of the reachability information in the overlay without the risk of causing problems in the basic IP connectivity the underlay provides.[7]

Link State in the Underlay

Link-state protocols, like BGP, are also widely implemented and understood. Every commercial routing stack and many open-source routing stacks—including an implementation of *Open Shortest Path First* (OSPF) or IS-IS—are mature, well tested, and widely deployed. However, most of these implementations are not optimized for use on DC fabrics. This section considers the positive aspects of using a link-state protocol on a DC fabric, some of the challenges operators face when deploying standard link-state protocols on DC fabrics, and realistic expectations for scale when using these unmodified implementations. The following sections address modified link-state protocols currently being designed and implemented, and the probable scaling characteristics of these implementations.

The first advantage link-state protocols have over BGP in DC fabrics is convergence speed—but the irony is link-state protocols are at their fastest where BGP is at its slowest, and vice versa. Link-state protocols are most challenged at scale during initial convergence because of the density of the topology through which flooding must take place. Considering the network in Figure 1, shown previously; when E1 originates a new *Link State Update* (LSU)—whether a *Label Switched Path* (LSP) fragment in IS-IS or a *Link-State Advertisement* (LSA) in *Open Shortest Path First* (OSPF)[11,12], it sends the update to every router in row D. Every router in row D, in turn, sends the LSU to every router in row E, which then sends the LSU to every router in row D. The number of copies each fabric device receives depends primarily on timing, but in topologies of around 2,600 fabric devices, each one was observed receiving more than 40 copies of each LSU. Nonetheless, unmodified link-state protocols converge at their worst as fast or faster than BGP up to some scale, where scale includes both the number of devices (nodes in the *Shortest Path Tree*, or SPT) and the number of reachable destinations. To what scale? The number will vary, but 1,000 (or more) fabric devices with a 100,000 reachable destinations are not unreasonable within a single flooding domain (or area in OSPF terms) based on prior large-scale deployments. Optimizations will increase these numbers somewhat—though to what degree depends on many factors.

Where link-state protocols converge much faster than BGP is when a reachable destination either moves from one place on the fabric to another or is disconnected from the fabric entirely. From the perspective of IS-IS, any reachable destination changes are just changes in leaf connectivity, meaning the destination can just be removed from the SPT without running *Shortest Path First* (SPF). This process is called a *partial SPF*; it is extremely fast and requires minimal processing on each of the fabric devices.

The second advantage link-state protocols have over BGP in DC fabrics is topology *visibility*. Link-state protocols require each device to maintain a full view of the topology, which must be synchronized with every other router in the network (or rather flooding domain); this process is called the *Link State Database* (LSDB). To obtain a copy of the LSDB, you need only to connect to one (or two, if you are concerned with resilience) router connected to the fabric. This kind of information is useful for traffic engineering and traffic steering. Further, periodic snapshots of the network topology from the perspective of the control plane can be a useful mine of telemetry information.

The first challenge for link-state protocols in the DC fabric is scaling, mainly related to flooding. We will consider several ways to reduce the number of LSUs each device receives in the following sections, so we don't consider them here. Another problem often cited in this area is the impression that link-state protocols can drop or fail to deliver LSUs—that flooding is periodic, rather than reliable, and the period is long enough to allow significant problems to develop. All link-state protocols, however, use reliable transport to deliver flooded packets. For instance, IS-IS tracks whether each neighbor has received an LSU through acknowledgments and retransmits LSUs until they are acknowledged. IS-IS can also send a description of the entire database periodically to ensure a neighbor's LSDB is correctly synchronized. OSPF has similar mechanisms.

Two other challenges link-state protocols face are scaling the number of reachable destinations and the time required to run the SPF algorithm used to calculate the set of loop-free paths. Faster processors combined with well-designed and well-tested implementations of SPF, along with optimizations such as partial SPF, have largely mitigated these concerns up to much larger scales than many engineers realize. Link-state protocols will never scale to the same levels as BGP, but they will scale enough to support a large proportion of the DC fabrics operators will build.

This article considers three proposed methods to control flooding designed to allow link-state protocols to support dense large-scale topologies. The first is *Distributed Optimized Flooding* (distoptflood), arguably the least complex of the three options. The second is a centralized flooding controller, and the third is *Routing in Fat Trees* (RIFT), which is essentially a modified link-state protocol designed specifically for spine-and-leaf fabrics.

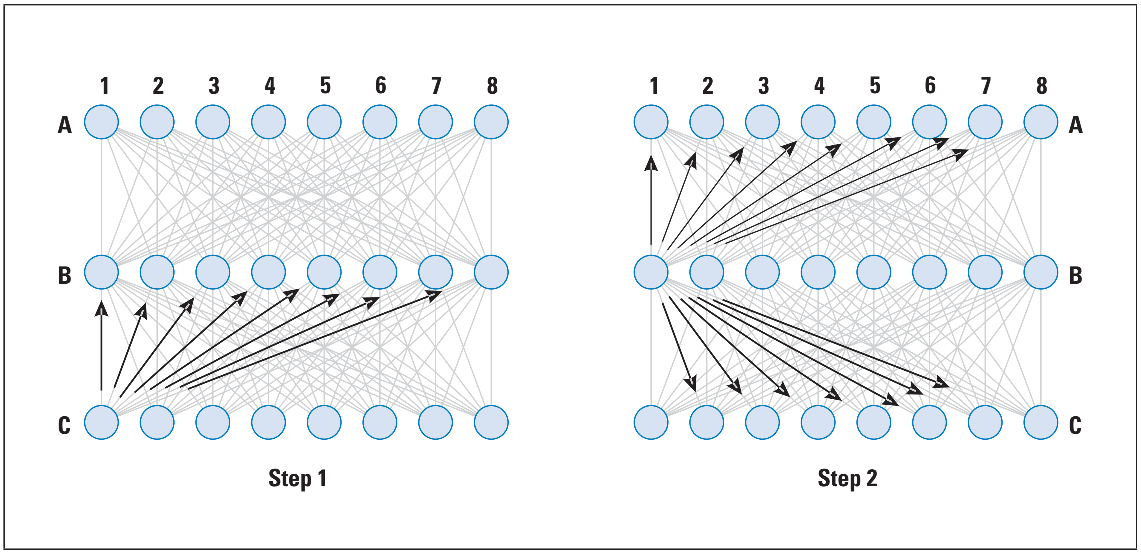
Distributed Optimized Flooding

Distoptflood outlines two optimizations to flooding, both of which work across all topologies and do not require a centralized controller of any kind. The first optimization is selecting a reduced set of reflooders[8] when flooding an LSP (or fragment—LSP is used interchangeably with LSPF fragment in these explanations) by doing the following:

- Set all link metrics to 1.
- Calculate the shortest path tree.
- Group nodes with a cost of 2 by directly connected neighbors (nodes reachable with a cost of 1) through which they are reachable.
- Select a set of directly connected neighbors that can reach all nodes with a cost of 2.
- Remove any directly connected neighbors that are on the shortest path towards the origin of the change.

Figure 2 illustrates the flooding optimizations in a Clos fabric, while Figure 3 illustrates flooding optimizations in a Butterfly fabric.

Figure 2: Distributed Optimized Flooding in a Clos Fabric



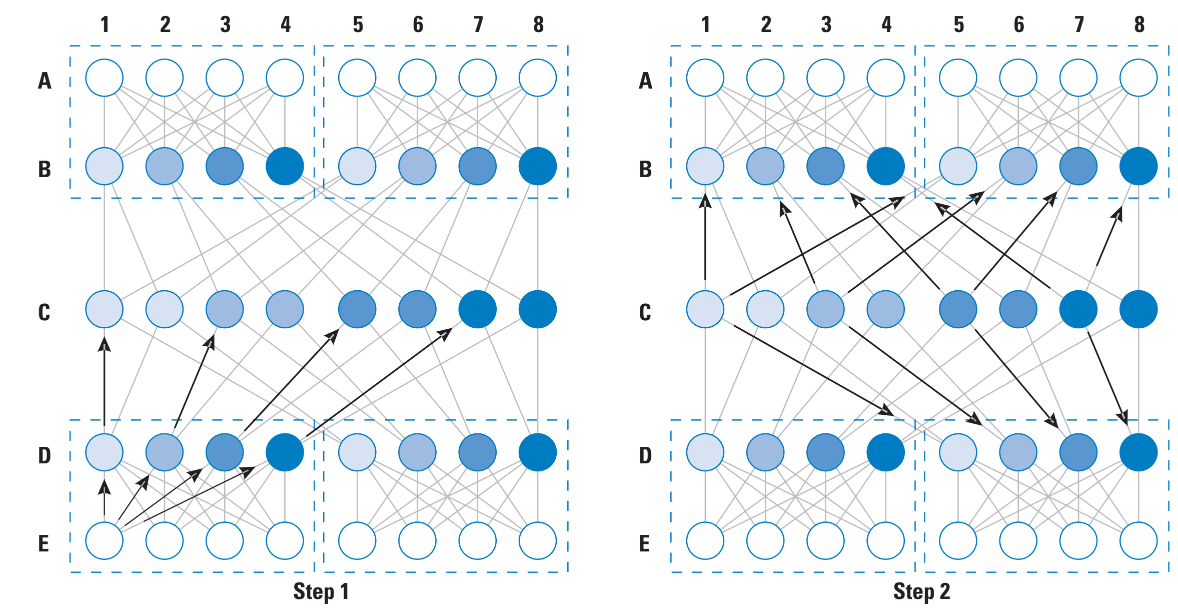
In the three-stage Clos (Figure 2), some change happens at C1; for instance, some network that was connected to C1 is disconnected. C1 calculates its two-hop neighborhood and determines it needs only to designate one of its neighbors, B1 through B8, as a reflooder. Let's assume it chooses B1 as the reflooder. C1 will send the LSP to C1 normally, and send the LSP to B2 through B8 using a link-local packet; these neighbors will receive the LSP and process it, but will not flood the changed LSP to their neighbors (they will not, in IS-IS terms, set their *Send-Receive* flag).

B1 will discover all of its neighbors can reach the same set of neighbors, and hence will select one connected neighbor as a reflooder; say B1 selects A1 as its reflooder. B1 will send the updated LSP to A1 through A8 and C2 through C8 using a link-local packet, so each of these routers will receive and process the change, but not reflood it. A1 will receive and process the update, but building its optimized flooding set will discover every one of its connected neighbors is on the shortest path towards the origin of the change, which is C1, so it will not reflood the update to any neighbors.

After about a second, each of the reflooders will send a *Complete Sequence Number Packet* (CSNP), which contains a description (or digest) of the local LSDB. If an IS notices a mismatch between its local LSDB and a neighbor's LSDB, it can send a *Partial Sequence Number Packet* (PSNP) requesting the retransmission of the missing information.

Figure 3 illustrates a slightly more complex five-stage spine-and-leaf fabric; while there is no "official" name for this configuration, it is often called a *Butterfly*.

Figure 3: Distributed Optimized Flooding in a Butterfly Fabric



Once again, let's assume a change occurs at E1, such as losing connectivity to a network (or reachable destination). In stage 1, E1 will build an LSP and calculate a set of reflooders that can reach its entire two-hop neighborhood—which is all of its neighbors in this case (D1 through D4). D1 through D4 will build a set of reflooders, which will include one of the two routers they are connected to in row C (C1 through C8). In stage 2, the selected reflooders in row C (C1, C3, C5, and C7) will determine a set of reflooders, which will be one spine router in each pod (such as A1, A5, and D5 for C1). The result: A1 through A8 and E5 through E8 will receive four copies of the changed LSP. None of the row A or row E routers will reflood the change because all their neighbors are on the shortest path back to E1, which originated the change. Depending on the timing of flooding, the number of copies of the changed LSP routers in rows A and E will likely be less than four.

A virtual testbed of around 2,600 routers configured as a butterfly showed this optimization decreased the number of LSPs each router received by a factor of 10 and doubled the initial convergence speed with more than 100,000 routes.

Because IS-IS runs over Ethernet natively and you can calculate the local system ID from an attached *Media Access Control* (MAC) (or Ethernet ID) address, you can run a modified IS-IS fabric with virtually no configuration on the fabric devices. You can assign locally calculated IPv6 addresses to the loopback address of each device, and use link-local IPv6 addresses to forward IPv6 traffic across the fabric. Forwarding IPv4 traffic would, of course, require an address plan and some form of automated configuration for loopback addresses. Fully autonomic configuration of this kind, however, can make troubleshooting issues and tracing flows through the fabric difficult. Therefore, current implementations do not include fully autonomic operations, so you must configure the system ID and the loopback address on each device.

A more controversial point is that using a control plane that runs natively at Layer 2 could improve security somewhat. A host that has been taken over or "pwned" by an attacker could not use the IP capabilities of the host to attack the operation of the fabric itself. Whether this feature results in an improvement in security is left to the reader (and operator!) to consider more deeply.

Centrally Calculated Optimal Flooding Trees

Centralized flooding management requires several modifications to link-state protocols, explained in "Dynamic Flooding on Dense Graphs"[2]—a framework describing the changes required rather than a specific implementation. Rather than approaching the problem of optimally flooding information through a dense topology using local calculations, you can calculate a *flooding leader*, which then distributes an optimal flooding tree to all nodes in the fabric. Individual nodes would normally flood only along the designated tree, and then "by request" to resolve any flooding issues or to add links temporarily without impacting the flooding topology.

Each flooding domain (area in OSPF terms) must have a flooding leader; the draft suggests OSPF can elect this leader in much the same way as the *Designated Router* (DR) or a *Designated Intermediate System* (DIS) in IS-IS, both of which are used to reduce the amount of flooding required to synchronize a set of routers connected to a single broadcast link. While a single leader per flooding domain is required, the draft suggests each flooding domain should have multiple candidates, so the failure of the flooding leader does not cause an outage. This setup would be similar to the way OSPF elects a *Backup DR* (BDR) first, promotes the BDR to the DR role, and then elects a new BDR. In this way, a new flooding leader can "listen in" and be ready to take over the role of flooding leader if failure occurs.

A new *Type Length Value* (TLV) is added to IS-IS to enable the election of an area leader. An IS advertising this TLV is considered in the area leader election on all devices in the flooding domain. Rather than specifying the algorithm used to elect the flooding leader, an algorithm field is used to indicate how the flooding leader should be elected. Perhaps the simplest algorithm would be to elect the device with the highest (or lowest) priority, as advertised in the new TLV, and select among multiple advertisers with the same priority using the system ID (in the case of IS-IS).

When elected, the flooding leader calculates an optimal flooding topology. The flooding leader does not need any special information here; it already has a full view of the topology of the flooding domain through the synchronization of LSDBs required by normal link-state protocol operation. The precise calculation used is not specific in the draft, but a simple one might be to just use the shortest path tree as calculated by the flooding leader as the optimal flooding tree. The flooding tree does not need to be optimal from every point in the topology; it is not used to forward traffic, only to reduce flooding. The flooding tree also does not need to be perfect. A single device receiving two copies of a flooded link-state change might be less than optimal, but it will not cause routing loops or other significant network problems. In the same way, if a device fails to receive some new link-state information, the result might be suboptimal traffic flow. The normal flooding processes in OSPF and IS-IS will eventually catch the error (generally on the order of seconds) and fix the problem. Some optimizations, such as choosing only one link from a set of parallel links, and handling multiple nodes connected to a shared multi-access link, are considered in some detail.

After the topology is calculated, it must be advertised to the network devices in the flooding domain. IS-IS advertises it using a new TLV that is similar to the way link-states are already advertised. Each TLV contains a series of system IDs through which the flooding path passes. Similar additions to the OSPF protocol are described as well.

What “Dynamic Flooding on Dense Graphs”[2] provides is a framework for a solution to flooding inefficiencies in link-state protocols, rather than a solution. In fact, you can advertise the use of distributed optimized flooding within a network by using the mechanisms in this draft. One specific algorithm for computing a dynamic flooding topology is described in “An Algorithm for Computing Dynamic Flooding Topologies”[3] in this way:

> “The proposed algorithm constructs a subgraph composed of small overlapping cycles. The base graph is denoted by G(V, E), where V is the set of all reachable nodes in this area, and E is the set of edges. The subgraph to be computed is denoted by G’({}, {}), which starts with an empty set of nodes and an empty set of edges.”

It is beyond the scope of this article to describe the precise way this algorithm operates or proposed alternatives.

RIFT

Routing in Fat Trees (RIFT) is a recent addition to this list, combining link-state and distance-vector concepts. Link-state-like operation is retained as information is transmitted up the fabric towards the ToF, while distance-vector-like operation carries reachability and topology information towards the edges of the fabric, the leaves.

Work on this new protocol started in IETF when the RIFT working group charter was approved in February 2018. The charter states:

> "The Routing in Fat Trees (RIFT) protocol addresses the demands of routing in Clos and Fat-Tree networks via a mixture of both link-state and distance-vector techniques colloquially described as 'link-state towards the spine and distance vector towards the leaves.' RIFT uses this hybrid approach to focus on networks with regular topologies with a high degree of connectivity, a defined directionality, and large scale."

The working group was chartered to create a protocol that will:

- Deal with automatic construction of fat-tree topologies based on detection of links.
- Minimize the amount of routing state held at each topology level.
- Automatically prune topology distribution exchanges to a sufficient subset of links.
- Support automatic disaggregation of prefixes on link and node failures to prevent black-holing and suboptimal routing.
- Allow traffic steering and rerouting policies.
- Provide mechanisms to synchronize a limited key-value data-store that can be used after protocol convergence.

According to the charter: "It is important that nodes participating in the protocol should need only very light configuration and should be able to join a network as leaf nodes simply by connecting to the network using the default configuration. The protocol must support IPv6 and should also support IPv4."

Basic Operations

As briefly described earlier, RIFT combines concepts from both link-state and distance-vector protocols. A *Topology Information Element* (TIE) is used to carry topology and reachability information; it is like an OSPF LSA or IS-IS LSP. Figure 4 illustrates the advertisement of reachability and topology information in RIFT.

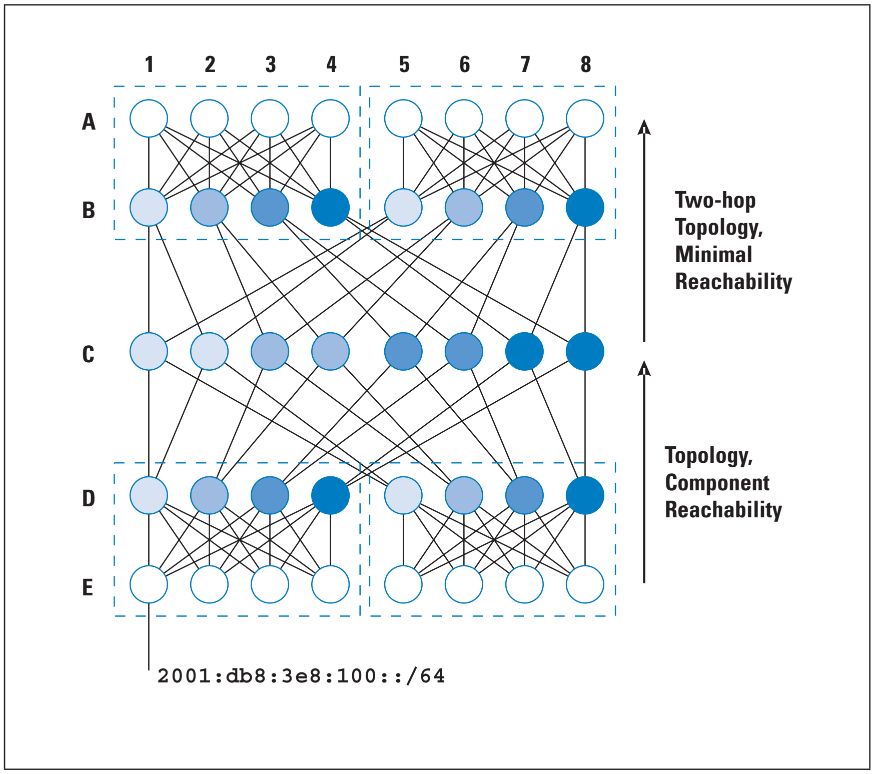
In Figure 4, E1, a ToR, advertises `2001:db8:3e8:100::/64` and its connections to D1–4 to D1–4. D1–4, in turn, refloods this information towards C1–8.

Rather than flooding the TIEs received from E1 back down the fabric, however, C1–8 advertises the minimal amount of reachability possible (normally this route would be a single default route) and their full set of neighbors to D5–8 and B1–8. When this process is completed:

- E2 will know about any locally connected destinations, four potential default routes from D1–4, and all the neighbors of D1–4 (the two-hop neighborhood for each connected neighbor).
- D5 will know about the neighbors and destinations connected to E5–8, the C1 two-hop neighborhood, the C2 two-hop neighborhood, and two possible default routes from C1 and C2.
- C4 will know about the destinations and neighbors reachable from all devices in rows A and E, and the neighbors connected to devices in rows B and D.

Using this information, the routers in row C can calculate a full SPT to discover the best path to each destination in the network. Routers in rows B and D will forward any traffic destined within the pod based on local information learned from the ToR switches, and all other traffic towards the default route learned from the ToF (row C). ToR switches will forward traffic using the default route learned from the spine stage above them, rows B and D.

Figure 4: RIFT Operation in a Butterfly Fabric


1
2
3
4
5
6
7
8
A
B
C
D
E
Two-hop Topology, Minimal Reachability
Topology, Component Reachability
2001:db8:3e8:100::/64

Automatic Disaggregation

In normal circumstances, devices other than those in the ToF stage rely on the default route to forward packets towards the ToF, while more specific routes are available to forward packets towards the ToR switches. What happens, however, if the C3—>D2 link fails? B2, B6, and D6 will continue forwarding traffic towards C3 based on the default route being advertised in the TIE flooded by C3, but C3 will no longer have a route by which it can reach `2001:db8:3e8:100::/64`—so this traffic will be dropped at C3.

To resolve this problem, RIFT can use the two-hop neighbors advertised to all routers to automatically determine when there is a failed link and push the required routes along with the default route down the fabric. In this case, C4 can determine D3 should have an adjacency with D2, but the adjacency does not exist. Because of the failed adjacency, C4 can flood reachability to `2001:db8:3e8:100::/64` alongside the default route it is already sending to all its neighbors. This more specific route will draw any traffic destined to the `100::/64` route, so C3 no longer receives this traffic. The default route will continue to draw traffic towards C3 for the other destinations it can still reach.

Other RIFT Features

When ToF fabric switches are configured, fabric devices running RIFT can compute their fabric location and largely self-configure (there are exceptions for devices requiring Layer 2 support and leaf nodes in the topology). This self-configuration includes the use of IPv6 *Neighbor Discovery* (ND)[13] to determine local IPv6 addresses, so no addressing plan or distribution protocols are required for pure IPv6 operation. If native IPv4 forwarding is required in the underlay, those addresses must be managed and configured in some way.

RIFT also offers the ability to perform unequal-cost load balancing from the ToR towards the ToF. Since each node has only the default route, and the stages closer to the ToF have more complete routing information, it is not possible for the ToR to cause a routing loop by choosing one possible path over another, or unequally sharing traffic along its available links.

Conclusion

BGP has been and will continue to be an important option for DC fabric underlays for many years to come. BGP may eventually offer some of the interesting features link-state protocols already offer, such as faster convergence and closer-to-autonomic deployment. On the other hand, some features of a link-state protocol, such as the ability to get a complete view of the entire topology from a single place—pulling a copy of the LSDB—are going to be very difficult to replicate in BGP, and the BGP convergence speed is always likely to lag behind a link-state protocol.

Table 1 summarizes many of the differences between the options outlined here.

Table 1: Differences Between Modified BGP, Modified IS-IS, and RIFT

Feature	BGP (Modified for DC Fabrics)	IS-IS (Modified for DC Fabrics)	RIFT
Peer Discovery	Partial	Yes	Yes
Automatic Tier Calculation	No	Potentially	Yes
Mis-Cabling Detection	No	Capability in progress	Yes
Fabric Addressing	Loopback address, peering; can be reduced with protocol modifications; can be automated	System ID, loopback address; can be automated or locally calculated	ToF state and others; can be automated
Aggregation; Default only on ToR and Below	Manually configured	No	Yes
Scales to Underlay Routing on Host	Yes	Depends on fabric size and implementation	Yes
High *Equal-Cost Multi-Path* (ECMP) Fanout Support	Yes	Yes	Yes
Unequal-Cost Load Sharing	Yes (in some implementations)	No	Yes
Full View of Topology	No	Yes	Yes (in the ToF)
Carry Opaque Configuration Data	No (can carry opaque information through Communities)	No (can carry opaque information through Tags)	Yes
Drain Node without Disruption	Yes	Yes	Yes
Automatic Disaggregation	No	No	Yes
Fast Convergence Speed	Partial (Depends on event type)	Yes	Yes
Security Includes Origin Validation and Replay Protection	Origin validation could be implemented, but heavy weight; no replay protection	No	Yes
Initial Implementation	Simple	Moderate	Complex
Overlay Support	Assumes single protocol (eBGP underlay, iBGP/eVPN overlay)	Assumes eVPN overlay	Supports eVPN overlay, can operate pure Layer 3 fabric with no overlay to the workload
Support for General Topologies (not just DC fabrics)	Yes	Yes	No

Link-state protocols offer a different set of tradeoffs than BGP does; operators would do well to consider the link-state options described here as strong alternatives to using BGP for DC fabrics underlays.

References and Further Reading

[0] Petr Lapukhov, Ariff Premji, and Jon Mitchell, "Use of BGP for Routing in Large-Scale Data Centers," RFC 7938, August 2016.

[1] Russ White, Shraddha Hegde, and Shawn Zandi, "IS-IS Optimal Distributed Flooding for Dense Topologies," Internet Draft, Work-in-Progress, September 2019, `draft-white-distoptflood-01`.

[2] Tony Li, Peter Psenak, Les Ginsberg, Huaimo Chen, Tony Przygienda, Dave Cooper, Luay Jalil, and Srinath Dontula, "Dynamic Flooding on Dense Graphs," Internet Draft, Work-in-Progress, November 2019, `draft-ietf-lsr-dynamic-flooding-04`.

[3] Sarah Chen and Tony Li, "An Algorithm for Computing Dynamic Flooding Topologies," Internet Draft, Work-in-Progress, March 2020, `draft-chen-lsr-dynamic-flooding-algorithm-00`.

[4] Alankar Sharma, Dmitry Afanasiev, Tony Przygienda, Bruno Rijsman, and Pascal Thubert, "RIFT: Routing in Fat Trees," Internet Draft, Work-in-Progress, March 2020, `draft-ietf-rift-rift-11`.

[5] Justin Meza, Tianyin Xu, Kaushik Veeraraghavan, and Onur Mutlu, "A Large Scale Study of Data Center Network Reliability." In *Proceedings of the Internet Measurement Conference 2018*, Association for Computing Machinery, 2018. `https://doi.org/10.1145/3278532.3278566`.

[6] "Folded" unfortunately has two distinct meanings in spine-and-leaf networks. The original spine-and-leaf design, the Clos, was considered unidirectional, in that circuit setup proceeded in one direction through the fabric, and the resulting fabrics were nonblocking. Using a spine-and-leaf for bidirectional packet-switched traffic "folds" the fabric. However, folding also means drawing the fabric with the topmost tier at the top of the diagram and all the leaves along the bottom of the diagram.

[7] Note that in some implementations of BGP, the iBGP and eBGP I/O paths are handled separately, making iBGP and eBGP either closer to, or fully, two separate failure domains. You should consider this point when determining which implementation to deploy in a DC fabric when BGP is used as the underlay protocol.

[8] The "first flood" to build an initial LSDB on which the two-hop neighborhood can be calculated is performed in the normal way; there is no optimization of this initial flood of topology information.

[9] Alvaro Retana and Russ White, "Optimizing Link-State Protocols for Data Center Networks," *The Internet Protocol Journal*, Volume 16, No. 2, June 2013.

[10] Yakov Rekhter, Susan Hares, and Tony Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006.

[11] John Moy, "OSPF Version 2," RFC 2328, April 1998.

[12] Dennis Ferguson, Acee Lindem, and John Moy, "OSPF for IPv6," RFC 5340, July 2008.

[13] William Allen Simpson, Thomas Narten, Erik Nordmark, and Hesham Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, September 2007.

[14] Wikipedia entry for "Clos network": **https://en.wikipedia.org/wiki/Clos_network**

RUSS WHITE began working with computers in the mid-1980s, and computer networks in 1990. He has co-authored more than forty software patents, participated in the development of several Internet standards, helped develop the CCDE and the CCAr, and worked in Internet governance with the Internet Society. Russ has a background covering a broad spectrum of topics, including radio frequency engineering and graphic design, and is an active student of philosophy and culture. Russ is a co-host of the *History of Networking* podcast, hosts the *Hedge* podcast, serves on the Routing Area Directorate at the IETF, co-chairs the BABEL working group, and serves on the Technical Services Council as a maintainer on the open-source *Free Range Routing* project. His most recent works are the book *Computer Networking Problems and Solutions*, *Network Disaggregation Fundamentals* video training, and *Abstraction in Computer Networks* video training. E-mail: **russ@riw.us**

MELCHIOR AELMANS is Lead Engineer Cloud Providers at Juniper Networks, where he has been working with many operators on the design, security, and evolution of their networks. He has over 15 years of experience in various operations, engineering, and sales engineering positions with cloud providers, data centers, and service providers. Before joining Juniper Networks, he worked with eBay, LGI, KPN, etc. Melchior enjoys evangelizing and discussing routing protocols, routing security, and internet routing and peering. He also participates in IETF and RIPE and is a board member at the NLNOG foundation. E-mail: **melchior@aelmans.eu**

So You Want to Sell Your IPv4 Address Block?

by David Strom

If your company owns a block of IPv4 addresses and is interested in selling it, or if your company wants to purchase additional addresses, now may be the best time to do so. As readers of *The Internet Protocol Journa*l (IPJ) are well aware, the number of available IPv4 addresses has been steadily dwindling, to the point now that many of the *Regional Internet Registries* (RIRs) are no longer assigning them. It may be a good time to look at the used-address marketplace. This arena could be a new corner of the Internet for you, so this article can help you understand what is going on and prepare you to do business in it.

For sellers, a good reason to sell address blocks is to make money and get some use out of an old corporate asset. If your company has acquired other businesses, particularly ones that have assets from the early Internet pioneers, chances are you might already have at least one range that is gathering dust, or is underused. Think of this idea of selling blocks as similar to how your company might decide to sell or release its unused real estate. "Many companies have millions of unused IP addresses," said Vincentas Grinius of Heficed, an address leasing vendor. "They have been holding on to them for future growth or to save as a strategic asset." Now might also be a good time to sell since prices are starting to level off, according to several brokers that I spoke to (of course, they have a vested self-interest), and the practice is becoming more accepted.

If you're a buyer, it is also a good time for you, as a way to extend the life of your enterprise IPv4 equipment for a few more years. It is particularly true if your business has resisted a full IPv6 deployment or you can't easily upgrade your legacy endpoints.

Until recently, the used-address marketplace hasn't had the best of reputations. Many of us imagine that getting a used-address block from a broker is like buying a cheap used car. Grinius told me that used addresses used to be thought of "as akin to *Hustler* magazine, something folks were ashamed of having in their possession." But things have gotten more legitimate: in addition to the used-car metaphor, you should also add the digital equivalence of a title insurance company and an accident reporting service like Carfax to establish more of a trusted exchange among buyer, broker, and seller.

Certainly the used-address market is thriving and quite competitive: now we have dozens of block brokers and at least three block lessors (IPv4 Market Group, Prefix Broker, and Heficed) that have solid business operations to help match buyers and sellers.

I have owned my own Class C block since 1993, and it seemed like an opportune time to sell it when IPJ's editor asked me to write about the used IPv4 marketplace.

So let's first review the history of the IPv4 address depletion and how RIRs work in terms of address allocation before we get into the specifics of how the broker/resale space works. Along the way I will offer my own comments about my experience in selling my own block, and what I learned that can help you decide whether you want to become a buyer, a seller, or a lessor of your own block.

Address Transfer Reference Library

Perhaps the best source of information about IPv4 address depletion, myths (such as changing out customer routers is easy and ISPs still have plenty of IPv4 addresses) and tools about IPv6 transition are available in the back issues of IPJ itself, including articles that Geoff Huston of the *Asia Pacific Network Information Centre* (APNIC) wrote. Following is a guide to the most useful pieces, in IPJ and elsewhere. Note that I shared my thoughts with Huston prior to publication, and have woven in some of his remarks in this summary.

- An IPJ June 2003[1] article reviews the early stages of IPv6 and includes some early myth busting by Huston, such as IPv6 has innately better security, *Quality of Service* (QoS), and mobility support. In one article, Huston says that "With a continuation of current policies it would appear that IPv4 address space will be available for many years yet." He was right, just perhaps not in the way that he originally intended. In a recent e-mail, Huston told me, "At the time there was a common expectation that the adoption of IPv6 was meant to complete before the IPv4 pool had exhausted itself."

- Four years later in an IPJ September 2007[2] article, Huston talks directly about the state of IPv4 address depletion, and has models that (accurately as it turned out) predicted full depletion by 2011. At that time, address exhaustion was pretty much inevitable. In the article, Huston stated that IPv6 didn't have a very compelling business case and that the use of *Network Address Translation* (NAT) in IPv4 is far easier, a claim you could still make today.

- An IPJ March 2011 special issue on the IPv6 transition[3] includes commentary on *World IPv6 Day* held June 2011 and a history of the address exhaustion of IPv4. "The stock of [IPv4] addresses is facing imminent depletion," Huston wrote in that issue. By then, APNIC had exhausted its IPv4 address pool. "Most of the actors in the Internet are unsure about what needs to be done [to make the v6 transition], from the largest of the service providers down to individual end users," he wrote. That issue is worth reviewing because it has a lot more helpful information about making the IPv6 transition.

- A December 2019 presentation by Huston about IPv6 is also worth reading[4] He discusses current pricing trends on block sales and predicts that by the time we run out of IPv4 addresses we will have outgrown IPv6: "We didn't need it back when it was first proposed and we still don't need it now."

Huston mentioned in a recent e-mail to me that it has been “nine years after the initial exhaustion point and IPv6 is still used by less than a quarter of the Internet and IPv4 remains the mainstay of the Internet. It was easier for the industry to change the entire architecture of the Internet than it was to universally adopt a new IP protocol.”

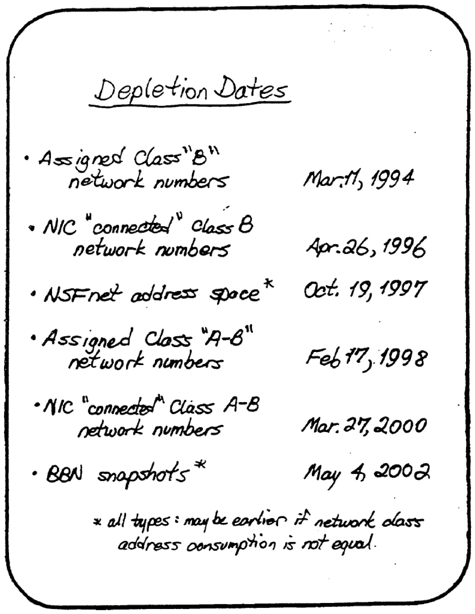
- A nice historical review of the development of RIRs is available in the December 2001 IPJ.[5] It covers *Classless Inter-Domain Routing* (CIDR), subnetting, and supernetting.
- A white paper from Eric Bais[6] has loads of practical advice for address transfer, written from the perspective of a broker in the *Réseaux IP Européens* (RIPE) region who both sells and leases blocks.

Historical Review

I first wrote about the depletion of the IPv4 address space when I was editor-in-chief of *Network Computing* magazine back in the early 1990s. Alas, that article is no longer accessible online. I remember it vividly because it got an amusing comment from my father, who never really understood technology but thought it would be funny if I were to leave my job and become an address broker. Needless to say, it was just a passing but prescient thought.

Back in these early days of the commercial Internet, Jon Postel personally and manually assigned IP address ranges. Usually he did it within moments of receiving an e-mail request, and that is how I got my /24 block. Obviously it didn’t scale after the Internet caught on. One of the first to sound the alarm was Frank Solensky, who published his predictions for various run-out dates in 1990 during the 18th meeting of the *Internet Engineering Task Force* (IETF).[7] See Figure 1.

Figure 1: Solensky’s Original Estimates of Address Exhaustion



Depletion Dates

• Assigned Class "B" network numbers	Mar. 11, 1994
• NIC "connected" Class B network numbers	Apr. 26, 1996
• NSFnet address space*	Oct. 19, 1997
• Assigned Class "A-B" network numbers	Feb 17, 1998
• NIC "connected" Class A-B network numbers	Mar. 27, 2000
• BBN snapshots*	May 4, 2002

* all types : may be earlier if network class address consumption is not equal.

The basic "Goldilocks" issue is that for the average business looking to get online, 250 addresses for a class C block is too little and 65,000 addresses for a class B block is too many. Numerous technical approaches have been proposed, including classless addressing (RFC 1918[8]), NAT, elimination of assigning static addresses to dial-up users, and changes to routing protocols. But the real solution was inventing IPv6 to increase the overall address space. During the early 1990s, the larger blocks of A and B ranges were already being rationed, given that Postel by then had previously assigned many of these blocks.

While the IPv4 addresses were being depleted, three of the RIRs were created through RFC 1366[9], modified by RFC 1466 in 1993[10], and further refined a few years later in RFC 2050[11]. Now there are five of them:

- *African Network Information Centre* (AFRINIC) serving Africa
- *Asia Pacific Network Information Centre* (APNIC) serving parts of Asia and the Pacific region
- *American Registry for Internet Numbers* (ARIN) serving North America and parts of the Caribbean
- *Latin America and Caribbean Network Information Centre* (LACNIC) serving Latin America and parts of the Caribbean
- *Réseaux IP Européens Network Coordination Centre* (RIPE NCC) serving Europe, parts of central Asia, and the Middle East

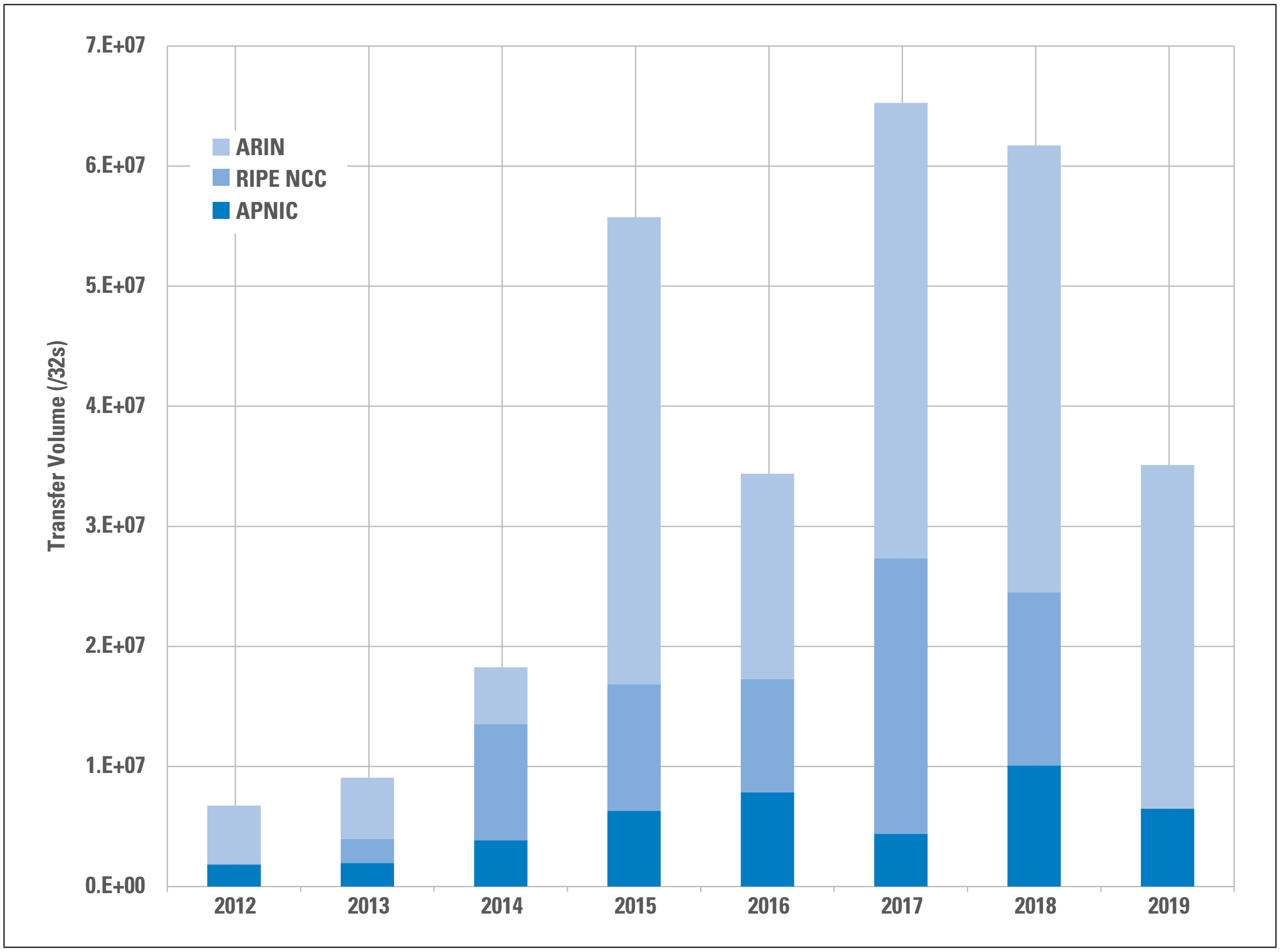
By February 2011, the last remaining common blocks of IPv4 addresses were fully allocated to the RIRs. In an article in IPJ, Raúl Echeberría, Chairman of the *Number Resource Organization* (NRO), the umbrella organization of the five RIRs, was quoted as saying, "It's only a matter of time before the RIRs and ISPs must start denying requests for IPv4 address space."[3] Today almost every block is assigned to some entity. AFRINIC has the most available, and APNIC has a few smaller blocks left. RIPE made its last /22 block assignment in November 2019.[12]

The Rise of the Used-Address Marketplace and RIR Supervision

Perhaps the origin event for the used-address market was when Microsoft purchased Nortel's inventory of more than 600,000 individual IPv4 addresses for US$7.5M in 2011. (Well, that isn't a completely accurate statement, but it does appear that Nortel's address pool was the main corporate asset.) Since then, tens of millions of addresses have been transferred[13] per year, as you can see in Figure 2.

In the last decade, the RIRs have played an increasing role in these transfers. In the references I have the direct links to the current transfer policies for each registry.[14] Note that some RIRs have more precision and transparency about their process, along with higher thresholds, than others to prove existing ownership of an address block.

Figure 2: Address transfers from 2020 statistics compiled by Geoff Huston



But this system wasn't perfect by any means: block ownership questions weren't easily resolved within a single registry, organization records were full of stale data or listed businesses that were no longer operating entities, and spammers could pollute address blocks by clouding any resale opportunities. Also, many address blocks (such as the one that I owned) pre-date the establishment of RIRs, what they call "legacy resources." How the RIRs deal with these assignments is a challenge, particularly as businesses are no longer around, and tracing the lineage from the original Postel assignment to a current stakeholder can involve some detective work. The question is, who should do the detecting? That isn't a simple question to answer, as you'll see.

Part of the problem was WHOIS itself, the primary domain and block ownership query tool. However, WHOIS is far from perfect. First, its responses differ depending on the data being queried, the RIR in charge of that block, and whether the block owner has provided accurate and up-to-date information or deliberately hidden these details.

But another part of the problem is that the Internet community has made changes to the display of information from WHOIS queries. Changes were necessary because of privacy concerns (from various changes to regulations around the world) and from spammers abusing WHOIS to drive legitimate business owners into hiding their details. I have placed the links to the different RIR WHOIS pages in the reference section if you want to compare them.[15]

If you were to examine my own /24 block before I began writing this article, you would see:

```
Organization: David Strom, Inc. (DAVIDS-3)
RegDate: 1993-05-21
Updated: 1996-04-18
```

The address used for my DAVIDS-3 organization is a New York corporation that is no longer in business. And the point of contact listed is an engineer at an ISP that I used to register the block that is also no longer in business. My challenge: I had to prove that the David Strom Inc. that did business in New York was the same David Strom Inc. that is now doing business in Missouri. Other than finding the plane ticket that I used in my move, I wasn't sure what else I could do to document the "asset transfer" that ARIN was going to eventually ask for.

Thus began my own journey to correct this information and get it ready for resale. The process involved spending a lot of time studying the various transfer webpages at ARIN, calling their transfer hotline several times for clarifications on their process, and paying a $300 transfer fee to start the process. ARIN staff promises a 48-hour turnaround to answer e-mails, and that can stretch out the time to prepare your block if you have a lot of back-and-forth interactions, as I did.

Enter the Block Broker

This discussion brings us to the modern era (say after 2012) and the IP block-broker marketplace. The goal was to try to make it easier for these transfers, and at the same time improve trust among all parties. As I said earlier, we now have many block brokers doing business. The broker's service (for either selling or leasing a block) is somewhat similar and involves these basic steps:

1. You need to register your business with the broker, a process that involves just answering a few basic questions and creating a login ID so you can interact with them via their various web-based forms and forums and e-mail.

2. Next, if you are a seller, you sign a mutual *Non-Disclosure Agreement* and then list your block that you want to sell. Some brokers have a variety of sales methods, including open and closed auctions and the ability to "buy now." If you are a buyer, you can start browsing the blocks that are available on the open auctions, and participate in the auction. If you have ever bought or sold any physical object via an online auction, you should be familiar with this process.

3. After you select a buyer for a particular block, you request the funds and place them in escrow, and then close the auction.
4. The broker's support team arranges for the transfer with the relevant RIR(s). As a buyer, you will then pay the fees directly to the RIR(s) for the transfer. Each RIR has a different way to calculate fees, ranging from free for RIPE to thousands of dollars, depending on the size of the block. (See Figure 3.)

Figure 3: The Different Fees Each RIR Charges for Transferring Addresses

RIR	Transfer Fee Amount
ARIN	$300 USD
RIPE	$0 USD
APNIC	20% of the annual fee for the # of IPv4 addresses being transferred
LACNIC	Initial payment of $200 USD Smaller than a /19 - $1,000 USD /19 and larger - $1,500 USD
AFRINIC	Smaller than /22 - $0 USD /22 to /20 - $1,750 USD /20 to /18 - $2,000 USD

5. Finally, the transaction closes and the block control and the funds released from escrow, minus any commission from the broker, are transferred to the buyer or leaser. Here is where things get interesting. The commissions aren't transparent: you have to get far enough down the process before you can find out what they are; the brokers set up the process this way deliberately so you can't shop around for lower fees. Still, there is a place for brokers, since "nothing is more frustrating than trying to get paid in a country of which you don't know the legal system nor have local representation. Using an escrow makes things easier for all parties involved," says Eric Bais of Prefix Broker.[6]

One other caveat for block leases: the lessor and lessee have a more intimate and longer-term relationship than if you are buying and selling the block outright, because ultimately the "landlord" business is still responsible for the reputation of the folks who are using your IP addresses. In other words, renting out your space also carries a certain risk to the lessor: just like rentals in the physical space, owners (or landlords) are responsible for their property. If you have a bad tenant who trashes your space, your reputation will suffer. This reality places a bigger burden of trust on the broker to ensure a proper tenant.

Three RIRs list brokers on their websites. They all have somewhat different contact information and number of brokers:

- APNIC has 22 listings[16], with contact names and phone and skype numbers.
- RIPE has 76 listings[17], with links to their contact webpages.
- ARIN has 29 listings[18], with contact names and phones and the date the broker registered with ARIN.

All of these RIRs try hard to indicate that their listings are not a recommendation, just awareness of their businesses. RIPE says its listing, for example, is of brokers who have agreed to conduct their business honorably, but no one checks on the brokers after they are listed to see if they have actually lived up to their promise. That is worth remembering. As the old saying goes, "on the Internet, no one knows if you are a dog."

If you are starting out in the used marketplace as I was, I recommend that you examine these RIR webpages carefully. Just having these lists of brokers is nice, but if you are going to sell or buy a used block you will find it frustrating to find the right broker for your situation. The biggest issue is that there are no fixed rules for buying, selling, and leasing used addresses. Unlike the used-car industry, there is no overall supervision or agreement on what constitutes the *quality* of an asset. As you can see from the five-step process cited previously, uncertainties and potential problems can arise at every step.

One other thing worth mentioning should be obvious but isn't: The only entities that can play are businesses. If you own a block as an individual, you will first have to transfer ownership to a business to proceed. You notice my ownership is my S-corporation (with my name; that helped me in the transfer process from my New York corporation to my Missouri corporation). Had I initially registered for my block as an individual, I might have had to work harder to prove my identity.

Important Caveats for the Transfer Process

Following are some of the complicating factors to watch out for as you begin your own transfer journey:

First, choosing whether to buy or lease a block can be tricky, and it depends on how many addresses you need and for what purpose. More details will follow, but you need to make this very basic decision before doing anything else, and often you won't have as much data as you might like.

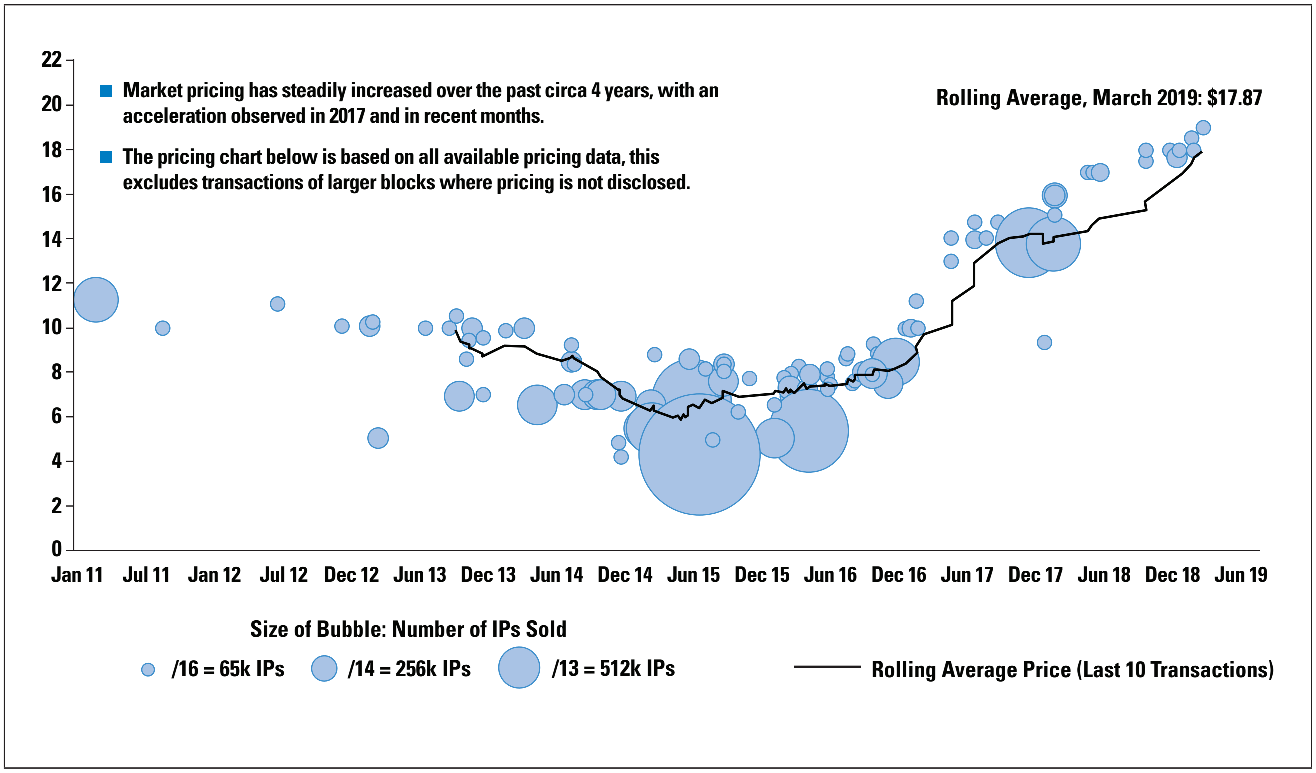
Unlike the used-car industry, there aren't any generally accepted practices or guides to making a tradeoff between buying and leasing. Of course, one aspect is the overall cost, and to estimate it you have to know your time horizon. If you are a buyer, do you need the block for a few years or a few months? Can you eventually migrate the endpoints to IPv6 using these addresses?

If you are a seller, do you want to dispose of the block and make a quick addition to boost your current year's balance sheet, or do you want to invest in a steady rental income over time? As a renter, you are also betting on a particular price curve over the terms of the lease that may or may not materialize. Now imagine that you are having this conversation with your Chief Financial Officer, who may or may not understand the various subtleties about the used-address marketplace.

You should base part of your choice of whether to rent or buy on the size of the block involved. Some brokers specialize in larger blocks and some won't sell or lease anything less than a /24, for example. "If you are selling a large block (say a /16 or larger) you would need to use a broker who can be an effective intermediary with the larger buyers," said Geoff Huston in an e-mail to me. Again, knowing that your broker has listed prior transactions can help you make a more informed decision. Not all brokers have pricing transparency, and many brokers are more circumspect about pricing.

IPv4.Global is one that does list their own prior auction sale data[19], for example. Another broker, IPv4 Market Group, has assembled the overall pricing chart shown in Figure 4 from March 2019[23]. There is no way to independently verify this information, but at least these examples show you how the market has evolved over the past decade.

Figure 4: IP Address Block Pricing Trends over Time



In early January 2020, /24s were selling at around US$20–24 per IP address, or US$5,000–6,000 for the entire block. Rental prices varied from 20 cents to US$1.20 per month per address, meaning at best a 2-year payback and at worst a 10-year payback when compared to sales. I decided to sell my block: I wanted the cash, and didn't like the idea of being a landlord of my block any more than I liked being a physical landlord of an apartment that I once owned. You'll also want to ensure that the RIR that is responsible for your block recognizes the broker you eventually choose.

Second, there is no guarantee that any of these brokers is reputable and will actually deliver the goods, or even if the RIR listings and contacts for the broker are still accurate. There is no easy way to vet their operations, or even agree on overall metrics to be used as part of the vetting process. Unless you know them personally, or know someone who does, chances are the names of the brokers on the RIR lists will require additional research for you to decide whom you should use to sell your block. You can look to see their registration data with ARIN, if ARIN controls your block.

One possible vetting strategy is to inquire how the broker is involved in the various Internet governance committees in your region, or at least examine their posted attendee lists. The hypothesis for this strategy is that broker representatives who attend IETF, RIR, and network operator meetings such as The *North American Network Operators' Group* (NANOG)[20] are more reliable than those that have never been to any of these meetings. (For example, PrefixBroker.com claims on their website that they helped author the RIPE transfer rules.)

IPv4 Market Group has a list of questions[21] to ask a potential broker, including if they will represent only one side of the transaction (most handle both buyer and seller) and if they have appropriate legal and insurance coverage. I found that a useful starting point.

Some brokers are also involved (either as other lines of business at their own company or as a subsidiary of a larger corporation) in other network- and Internet-related businesses, such as hosting and cloud services, while others operate in real estate development and intellectual property litigation. That may be relevant, or it could cloud your evaluation if the quality of these other businesses differs from that of the brokerage.

One of the reasons I went with IPv4.Global/Heficed was because of their transparency in terms of showing me the active auctions and past sales of their blocks right on their web homepage, and they e-mailed me periodically with the active and closed auctions.

Third is how you vet the other party in your transaction. In other words, if you are a seller, what process do you use to know your buyer, and vice-versa?

You might want to consider longer-term contracts for rentals (such as 3 years) for stability and also to minimize the movement of their tenants. “I would be somewhat worried if the broker did not undertake some diligence steps directly to validate the credentials of the seller,” said Huston in an e-mail to me.

The final part of the transfer process is to understand the condition of the actual address block itself. There is no guarantee that a used block isn’t tainted with spammers or used for other less-than-legal activities. “There are no established standards of conduct, little transparency, and even less accountability,” wrote Marc Lindsey in 2018 for a blog post on *CircleID*.[13] “Many participants in the market struggle to define, from a legal perspective, what is being bought and sold.” He also has several suggestions on vetting the other party in the transaction that are worth reviewing.

Most brokers will state that they examine prior ownership of their blocks to ensure they are spam-free and to eliminate the potential of being used for other shady dealings. The trick is understanding what tools they use to convince you of this claim. For example, some brokers require you to check the blacklists (such as those maintained at Cisco Talos, Hetrixtools.com, and IP-score.com) on your own to ensure that your block isn’t listed there. IPv4 Market Group offers a blacklist cleaning service[22] that examines 90 blacklists. While charges vary, to give you an idea, they quoted me $2,000 as part of their selling services for my /24 block. IPv4.Global checks 20 different blacklists as part of their services.

However, identifying whether a block is on a blacklist and removing it from a list are two different matters. If it is listed, you will have to work on removal from the blacklists before you can lease it. According to Geoff, “Once an address is blacklisted it’s exceptionally hard to get it unlisted.” None of the brokers will give you a firm price on cleansing a block, because it depends on how many blacklists it appears on.

So what happened to my sale? It took 10 days to auction off my block. I worked with ARIN to transfer my ownership to my current corporation, and paid them a second fee of $125 for dealing with my legacy ownership. I then worked with my broker to finalize the sale. The overall elapsed time from beginning to end was 1 month, including about a week of elapsed time to conduct the initial research and select the broker.

Summary

If all that seems like a lot of work to you, then perhaps you just want to steer clear of the used marketplace for now. But if you like the challenge of doing the research, you could be a hero at your company for taking this task on. Expect the entire process to take several months from start to finish, allowing for time to get your ownership in order (if you have a legacy block), navigate the legal and other corporate approvals, research your broker, and then actually execute the transaction.

References and Further Reading

[0] Various authors, *ConneXions—The Interoperability Report*, Volume 8, No. 5, May 1994, Special Issue: IP: The Next Generation, available from The Charles Babbage Institute: `http://www.cbi.umn.edu/hostedpublications/Connexions/index.html`

[1] Geoff Huston, "Opinion: The Mythology of IPv6," *The Internet Protocol Journal*, Volume 6, No. 2, June 2003.

[2] Geoff Huston, "IPv4 Address Depletion and Transition to IPv6," *The Internet Protocol Journal*, Volume 10, No. 3, September 2007.

[3] Various authors, *The Internet Protocol Journal*, Volume 14, No. 1, March 2011. This special issue is devoted entirely to the addressing and transitioning topics.

[4] Geoff Huston, presentation slides about current issues with IP addressing, December 2019. `https://www.potaroo.net/presentations/2019-12-11-kismet-addresses.pdf`

[5] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal*, Volume 4, No. 4, December 2001.

[6] Eric Bais, "Transferring IPv4 Resources in the RIPE region," June 2016. An ebook published by Prefix Broker. `https://www.prefixbroker.com/ebook/`

[7] Frank Solensky, Proceedings of the 18th IETF, 1990. `https://www.ietf.org/proceedings/18.pdf` (see p. 67 of the PDF for his original predictions of address exhaustion)

[8] Daniel Karrenberg, Yakov Rekhter, Eliot Lear, and Geert Jan de Groot, "Address Allocation for Private Internets," RFC 1918, February 1996.

[9] Elise Gerich, "Guidelines for Management of IP Address Space," RFC 1366, October 1992.

[10] Elise Gerich, "Guidelines for Management of IP Address Space," RFC 1466, May 1993.

[11] Kim Hubbard, Jon Postel, Mark Kosters, Daniel Karrenberg, and David Conrad, "Internet Registry IP Allocation Guidelines," RFC 2050, November 1996.

[12] RIPE press release, November 2019, "The RIPE NCC Has Run Out of IPv4 Addresses," `https://www.ripe.net/publications/news/about-ripe-ncc-and-ripe/the-ripe-ncc-has-run-out-of-ipv4-addresses`

[13] Marc Lindsey, *CircleID* blog post July 2018. "An Insider's Guide to the IPv4 Market – Updated," `http://www.circleid.com/posts/20180710_an_insiders_guide_to_the_ipv4_market_updated/`

[14] Here are the links to the RIR webpages regarding their rules for transferring resources:

`https://www.apnic.net/manage-ip/manage-resources/transfer-resources`

`https://www.ripe.net/manage-ips-and-asns/resource-transfers-and-mergers`

`https://afrinic.net/resources/transfers`

`https://www.arin.net/resources/registry/transfers`

`https://www.lacnic.net/1019/2/lacnic/resources-transference`

[15] Here are the links to query the WHOIS resources at each RIR:

AFRINIC Database:
`https://www.afrinic.net/whois-web/public/query`

APNIC Database:
`https://wq.apnic.net/apnic-bin/whois.pl`

ARIN Database:
`https://whois.arin.net/`

LACNIC Database:
`https://lacnic.net/cgi-bin/lacnic/whois`

RIPE Database:
`https://www.ripe.net/manage-ips-and-asns/db`

[16] APNIC, Registered IPv4 brokers: `https://www.apnic.net/manage-ip/manage-resources/transfer-resources/transfer-facilitators/`

[17] RIPE, Brokers: `https://www.ripe.net/manage-ips-and-asns/resource-transfers-and-mergers/brokers`

[18] ARIN Registered Transfer facilities: `https://www.arin.net/resources/registry/transfers/stls/registered_facilitators/`

[19] IPv4.Global, prior auction pricing data: `https://auctions.ipv4.global/prior-sales`

[20] NANOG, attendees list of meeting #77: `https://events.nanog.org/events/nanog-77/attendees-15-13b224d66c30422494a9627a6dcb6c94.aspx`

[21] IPv4 Market Group, "Approved IPv4 Address Facilitator for Your IPv4 Needs, a Guide to Questions You Might Want to Ask Your Broker,"
`https://ipv4marketgroup.com/ipv4-market-group/`

[22] IPv4 Market Group, blacklist removal service.
`https://ipv4marketgroup.com/broker-services/ipv4-blacklist-removal/`

[23] IPv4 Market Group, "IPv4 Price Trends,"
`https://ipv4marketgroup.com/ipv4-price-trends/`

[24] Prefix Broker: `https://www.prefixbroker.com/`

[25] Heficed: `https://www.heficed.com/`

[26] Richard Jimmerson, "On the 'Misuse' of the Internet Number Resource Transfer Market," Team ARIN Blog, August 26, 2020.
`https://teamarin.net/2020/08/26/on-the-misuse-of-the-internet-number-resource-transfer-market/`

DAVID STROM has written several articles for IPJ, most recently on fileless malware in 2018. He was the founding editor-in-chief for *Network Computing* (USA) magazine and ran overall editorial operations for Tom's Hardware.com. He is the author of two books on computing, including one as co-author with Marshall T. Rose on Internet messaging. He lives in St. Louis and can be reached at: `david@strom.com` or Twitter `@dstrom`.

In Memoriam: Yngvar Lundh

by Ole Jacobsen, The Internet Protocol Journal

Yngvar Gundro Lundh (March 19, 1932 – August 15, 2020) was my friend, mentor, and boss at the *Norwegian Defence Research Establishment* (NDRE). I first met Yngvar around 1976 when I was still in high school working on a report about computers and society. I worked at NDRE in Yngvar's micro-computer group through my military service and later during summer vacations while at university. Yngvar was the person who introduced me to the wonders of computers, and most of all to networking. NDRE had access to the first ARPANET connection outside of the United States. It was through this link (a *TeleType* connected to the NORSAR-TIP itself connected at 9.6 kbit/s to the ARPANET via satellite) that I met many friends in the US, ultimately leading to my employment at the Network Information Center at SRI International in 1984.

Yngvar played a major role in fostering technology development in Norway through his work at NDRE, as professor of informatics at the University of Oslo, as chief engineer at Norwegian Telecom, and as consultant on a variety of projects, including the first commercial electronic mail system in Norway.

Photo: Gisle Hannemyr CC BY-SA 3.0

His group designed and built Norway's first transistor-based computer, SAM, which you can see at Norsk Teknisk Museum in Oslo.

He was perhaps best known for his early work with the ARPANET and SATNET at NDRE. Yngvar Lundh and Pål Spilling were largely responsible for getting Norway connected to the Internet in the early 1980s.[1,2,3,4]

I fondly remember Yngvar as a patient teacher, generous with his time and always willing to help with projects large and small. He played a major role in my university and career path, and I will very much miss his guidance and inspiration.

Yngvar had many hobbies, including gardening, bee keeping, wood working, ham radio (LA72C), and above all, sailing. After retirement, he moved from Skedsmokorset near Oslo to Tolvsrød near the coastal town of Tønsberg, allowing him easy access to his sailboat.

References

[1] Yngvar Lundh, “A Slice of Norway’s Computing History,” *IEEE Xplore*, April-June 2018.
`https://ieeexplore.ieee.org/document/8415734`

[2] Wikipedia article on Internet Pioneers:
`https://en.wikipedia.org/wiki/List_of_Internet_pioneers`

[3] Pål Spilling and Yngvar Lundh, “Features of the Internet History, The Norwegian Contribution to the Development,” *Telektronikk* 3.2004. Available from:
`https://www.usit.uio.no/om/organisasjon/sst/stab/ansatte/bness/tilkoplet/web/7/src/pal-spilling-yngvar-lundh-features-of-the-internet-history.pdf`

[4] Wikipedia article on Pål Spilling:
`https://en.wikipedia.org/wiki/P%C3%A5l_Spilling`

[5] Wikipedia article on Yngvar Lundh (in Norwegian):
`https://no.wikipedia.org/wiki/Yngvar_Lundh`

[6] Norwegian Defence Research Establishment:
`https://www.ffi.no/en`

[7] Dag Andreassen, “Internett med norske pionerer,” Norsk Teknisk Museum,
`https://www.tekniskmuseum.no/21-nyheter/354-internett-med-norske-pionerer`

[8] Tor Sverre Lande, “Nekrolog: Yngvar Lundh,” *Aftenposten*, August 27, 2020.
`https://www.aftenposten.no/personalia/i/1ngy8e/nekrolog-yngvar-lundh`

[9] Yngvar Lundh, *Konstruksjon av integrerte kretser*, Universitetsforlaget, 1983, ISBN13 9788200066910.

OLE J. JACOBSEN is the Editor and Publisher of *The Internet Protocol Journal*, a quarterly publication on all aspects of Internet technology. He has been active in the computer networking field since 1976, when he joined the Norwegian Defence Research Establishment, an early ARPANET site. Ole holds a B.Sc. in Electrical Engineering and Computing Science from the University of Newcastle upon Tyne, England. He serves on the board of the *Asia Pacific Network Operators Group* (APNOG), which hosts the annual *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) conference, and has served on several ICANN and IETF nomination committees. In his spare time, Ole organizes pipe-organ concerts and demonstration events. E-mail: `ole@protocoljournal.org`

Book Review

Transforming Information Security

Transforming Information Security: Optimizing Five Concurrent Trends to Reduce Resource Drain, by Kathleen Moriarty, ISBN-13 978-1839099311, Emerald Publishing Limited, July 2020.

When I was asked to write a short review about Kathleen Moriarty's book, I took a copy with me on my summer holiday. I usually try to stay away from work-related literature during vacation, but in this case it was well worth it. With some 200 pages packed with facts and information, this book requires a bit of concentration and focus. But you get a lot in return for the effort.

With her extensive background and expertise in security, Kathleen analyses five trends in the current security debate: End-to-End Encryption, Strong Session Encryption, The Evolution of the Transport Protocol Stack, Data-Centric Security, and More User Control.

With these trends in mind, Kathleen comes to the conclusion that we need a fundamentally different approach to network and information security. She promotes a more manageable system with a minimised, secure operating system and layered or hosted (authorised) applications on top of it. Vendors need to take more responsibility managing vulnerability, and should enable automated updates that users can trust.

Security has become increasingly complex and requires specialised knowledge and expertise. There is already a huge shortage of security practitioners. Training more people and buying additional security tools and products is not going to scale. It is increasingly challenging to secure our networks and keep them manageable at the same time. Only wide-scale adoption of end-to-end encryption, increased capabilities of the end-points, and a change of network architecture and security practices will help in the mid and end terms.

But Kathleen doesn't leave it at these high-level statements. The book is full of practical tips and provides a wide range of *Internet Engineering Task Force* (IETF) standards, guidelines, and suggested security frameworks that IT and security staff can find useful—the list of references in itself is very useful and encourages further reading.

Kathleen walks us through various aspects of information security: from threat detection and prevention, to the use of security control frameworks, to the need for more automation and the importance of sharing information with peers in the network and security community.

She also provides an overview of many standards and protocols such as IPv6, *Quick UDP Internet Connection* (QUIC), *Manufacturer Usage Description* (MUD), routing overlay protocols, *DNS over Hypertext Transfer Protocol Secure* (DoH), and *DNS over TLS* (DoT) to name only a few, and explains their relevance in the overall security landscape.

Some sentences are packed with information, and it is worth it to read them twice.

The book provides a peek into the hopefully not-too-distant future where applying a more holistic view on security, automation, and sharing relevant information will benefit the networking and security community.

—*Mirjam Kühne,* **mir@zu-hause.nl**

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**

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 several 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. Make sure that *both* your postal and e-mail addresses are up-to-date since these are the only methods by which we can contact you. If you see the words "Invalid E-mail" on your copy this means that we have been unable to contact you through the e-mail address on file. If this is the case, please contact us at **ipj@protocoljournal.org** with your new information. The subscription portal is located here: **https://www.ipjsubscription.org/**

Fragments

Announcement Regarding IPv4 Address Block 43/8

As many people know, I have dedicated much of my career to the development of research networks and network technologies in Japan and Asia. This included the WIDE (*Widely Integrated Distributed Environment*) Project, founded in 1985 for computer networking Research and Development. In the early days of the WIDE Project, we were aware of the exciting advent of the Internet, and I was often in contact with Jon Postel and other Internet pioneers, about how it could be brought to Asia.

In the late 1980s, I recognized the Internet's importance in the world and in Asia. I requested a number of early IPv4 Class B assignments (/16s), directly from Jon Postel as NIC function delegation trial, as well as Class Cs and a Class A, for use by research networks in Japan. Since then, I have been administrating the Class A assignment, 43/8, to assist in the long-term development of the Internet in the Asia Pacific region.

In the early 1990s, I helped to establish the *Asia Pacific Network Information Centre* (APNIC) from the *Japan Network Information Center* (JPNIC), to provide continuing allocations of IPv4 address space for our region, Asia Pacific, at a time when the Internet was growing very quickly. APNIC launched in 1993 and has been very successful in managing IPv4 address space since then.

Since 1992, I continued to lead the WIDE project, which was then dedicated to the development and promotion of IPv6. Some of the 43/8 address space was used for this purpose, to assist Japanese networks with renumbering in their transition to IPv6. Some of this space, a /11 in total, was allocated by APNIC to participants in that project, and the rest retained by the WIDE project for other R&D activities.

The deployment of IPv6 has been slower than expected, but I'm very happy that finally, IPv6 is in full production around the Internet, and used by around 25% of Internet users globally. It's clear now that IPv6 will succeed and that the Internet will be greatly improved as the transition continues into the future.

IPv4 has a continuing role on the Internet, but a relatively short-term role, as IPv6 adoption increases. Therefore, IPv4 address space has a current value, but a value that will reduce and disappear over the next 10 years or so. While I have not been an active supporter of the commercialization of IP addresses, the fact is that a market for IPv4 addresses exists and the APNIC community has remained neutral by developing a proper policy framework for market transfers.

In considering the future of 43/8 I have again considered how it may be best used for its original purpose. After careful consideration, I have taken a decision to release this address block, for the purpose toward healthy development of today's Internet services and toward supporting Internet development in the AP region. This is possible by making it available on the IPv4 address market. This is an opportunity to produce a capital asset, with a significant impact on Internet development, if used well and carefully. It is an opportunity that exists today and might not be repeated at any point in the Internet's future.

As I mentioned, APNIC has now been established for 27 years, and it has performed a critical and successful role. APNIC has served very well as the *Regional Internet Registry* for our region, and it has had a great impact in the development of the Internet in our region. With the establishment of the *APNIC Foundation* in 2016, it's clear that APNIC is committed to the continuation and expansion of that good work.

Recognising APNIC's role and its successes, I have asked APNIC to receive a transfer of the unallocated portion of 43/8, on two conditions: that the block will be placed on the IPv4 address market for those who still need IPv4 addresses, and that the proceeds be used in support of Internet development in our region. I am grateful that the APNIC Executive Council has accepted this offer and is now proceeding accordingly, with the establishment of a charitable trust the *Asia Pacific Internet Development Trust*, (APIDT) to take responsibility for this asset and its disposal on the IPv4 address market.

I will remain closely involved, personally and through the WIDE Project, in the management of the Trust, and in its support of Internet development in our region, primarily through the APNIC Foundation. I am very happy to have taken this step and am looking forward to the results in the coming years and decades. I thank everyone involved in this process.

—Jun Murai, Founder, WIDE Project, March 25, 2020

For further information, contact **secretariat@wide.ad.jp**

WIDE Project: **http://www.wide.ad.jp/**

APNIC: **https://www.apnic.net/**

APIDT: **http://www.apidt.org/**

APNIC Foundation: **https://apnic.foundation/**

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
Finn Arildsen
Tim Armstrong
Richard Artes
Michael Aschwanden
David Atkins
Jac Backus
Jaime Badua
Bent Bagger
Eric Baker
Santosh Balagopalan
Michael Bazarewsky
David Belson
Hidde Beumer
Pier Paolo Biagi
Tyson Blanchard
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
Chad Burnham
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
Guido Coenders
Brad Clark
Narelle Clark
Joseph Connolly
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
Dmitriy Dudko
Andrew Dul
Joan Marc Riera Duocastella
Pedro Duque
Holger Durer
Mark Eanes
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
Michael Fiumano
The Flirble Organisation
Gary Ford
Jean-Pierre Forcioli
Susan Forney
Christopher Forsyth
Andrew Fox
Craig Fox
Fausto Franceschini
Valerie Fronczak
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
Ron Goodheart
Octavio Alfageme Gorostiaga
Barry Greene
Jeffrey Greene
Richard Gregor
Martijn Groenleer
Geert Jan de Groot
Christopher Guemez
Gulf Coast Shots
Sheryll de Guzman
Rex Hale
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
Asbjørn Højmark
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
David Jump
Merike Kaeo
Andrew Kaiser
Christos Karayiannis
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
Antonin Kral
Robert Krejčí
Mathias Körber
John Kristoff
Terje Krogdahl
Bobby Krupczak
Murray Kucherawy
Warren Kumari
George Kuo
Dirk Kurfuerst
Darrell Lack
Andrew Lamb
Richard Lamb
Yan Landriault
Edwin Lang
Sig Lange
Markus Langenmair
Fred Langham
Tracy LaQuey Parker
Rick van Leeuwen
Simon Leinen
Robert Lewis
Christian Liberale
Martin Lillepuu
Roger Lindholm
Link Light Networks
Sergio Loreti
Eric Louie
Guillermo a Loyola
Hannes Lubich
Dan Lynch
Sanya Madan
Miroslav Madić
Alexis Madriz

Carl Malamud
Jonathan Maldonado
Michael Malik
Tarmo Mamers
Yogesh Mangar
Bill Manning
Harold March
Vincent Marchand
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
Carsten Melberg
Kevin Menezes
Bart Jan Menkveld
Sean Mentzer
William Mills
David Millsom
Desiree Miloshevic
Joost van der Minnen
Thomas Mino
Rob Minshall
Wijnand Modderman
Mohammad Moghaddas
Roberto Montoya
Charles Monson
Andrea Montefusco
Fernando Montenegro
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
Tim O’Brien
Mike O’Connor
Mike O’Dell
John O’Neill
Jim Oplotnik
Packet Consulting Limited
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
Fernando Robayo
Gregory Robinson
Ron Rockrohr
Carlos Rodrigues
Magnus Romedahl
Lex Van Roon
Alessandra Rosi
David Ross
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
Philip Schneck
Dan Schrenk
Richard Schultz
Timothy Schwab
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
Eric Smith
Mark Smith
Craig Snell
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
David Strom
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
Phillip Yialeloglou
Janko Zavernik
Muhammad Ziad Ziayuddin
Jose Zumalave
Romeo Zwart
Bernd Zeimetz
廖 明沂.

Follow us on Twitter and Facebook

`@protocoljournal`

`https://www.facebook.com/newipj`

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`

This publication is distributed on an "as-is" basis, without warranty of any kind either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This publication could contain technical inaccuracies or typographical errors. Later issues may modify or update information provided in this issue. Neither the publisher nor any contributor shall have any liability to any person for any loss or damage caused directly or indirectly by the information contained herein.

Supporters and Sponsors

Supporters

Internet Society

CISCO

Diamond Sponsors

Google

Ruby Sponsors

ICANN

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

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

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
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

Printed in the USA on recycled paper.